OIPE AUG 0 1 2005 PATENT & TRADEMARK OFFICE

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to:

On 7/28/05

TOWNSEND and TOWNSEND and CREW LLP

By: [signature]

PATENT
Attorney Docket No.: 16869B-098300US
Client Ref. No.: HAL-ID 289
Hitachi Ref. No.: 340301715US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Yuichi Yagawa

Application No.: 10/806,998

Filed: March 24, 2004

For: Distributed Data Management System

Customer No.: 20350

Confirmation No. 2821

Examiner: Unassigned

Technology Center/Art Unit: 2171

PETITION TO MAKE SPECIAL FOR NEW APPLICATION PURSUANT TO 37 C.F.R. § 1.102(d) & M.P.E.P. § 708.02, Item VIII, ACCELERATED EXAMINATION

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Submitted herewith is a petition to make special the above-identified application in accordance with MPEP § 708.02, Item VIII, accelerated examination. The application has not received any examination by the Examiner.

(A) The Commissioner is authorized to charge the petition fee of $130 under 37 C.F.R. § 1.17(h), and any additional fees that may be associated with this petition may be charged to Deposit Account No. 20-1430.

(B) All the claims are believed to be directed to a single invention. If the examiner determines that all the claims presented are not obviously directed to a single invention, then Applicant will make an election without traverse as a prerequisite to the grant of special status where the specific grouping of claims will be determined by the examiner.

08/02/2005 EFLORES 00000118 201430 10806998
01 FC:1464 130.00 DA

(C) A pre-examination search was performed by an independent patent search firm. The pre-examination search includes a classification search, a computer database search, and a keyword search. The classification search covered Class **360**, subclass 48; Class, **707**, subclasses 200, 201, and 204; Class, **709**, subclasses 203, 217, and 218; Class **710**, subclass 74; and Class **711**, subclasses 112, 154, 161, and 162. Additionally, a keyword search was performed on the USPTO full-text database, including published applications. The following references were identified in the search report:

(1) U.S. Patent Nos.:

5,790,886	Allen
6,035,351	Billings et al.

(2) U.S. Patent Application Publication Nos.:

2002/0065835	Fujisaki
2002/0147734	Shoup et al.
2002/0174306	Gajjar et al.
2003/0229637	Baxter et al.
2004/0039891	Leung et al.

(3) Foreign Publication Nos.:

EP 0617373	Burket et al.
GB 2367163	Martin et al.

(D) The above references are enclosed herewith, collectively as Exhibit A.

(E) Set forth below is a detailed discussion of the references, pointing out with particularity how the claimed subject matter recited in the claims, amended according to the preliminary amendment filed herewith, is distinguishable over the references.

Claimed Subject Matter of the Present Invention

There are six independent claims among the 28 claims that are pending in the instant application.

Independent **claim 1** relates to a method for distributing data among data storage systems. The method includes obtaining selection criteria. Profile information based on the content of a data object is produced. The data object is selectively copied from its location in a

first data storage system to a second data storage system based on the selection criteria and on the profile information.

Independent **claim 12** relates to a data storage system comprising data servers, each having a client interface, a data storage interface, and a data processing component. The data processing component produces profile information comprising information associated with content of a data object. The data processing component compares selection criteria with the profile information, where the selection criteria are associated with a second data server and determine whether the data object is copied to the second data server. The data processing component performs the copying depending on the outcome of the comparison.

Independent **claim 18** relates to a method for distributing data among data storage systems. The method includes obtaining selection criteria and storing the selection criteria on a first data storage system. Profile information based on the content of a data object stored in the first data storage system is produced. The data object is selectively copied from its location in the first data storage system to a second data storage system based on the selection criteria and on the profile information.

Independent **claim 20** relates to a data system comprising plural data centers and plural client systems. Each data center includes a data storage component, a file server, a replicator component, a receiver component, and file selection criteria. The replicator component receives selection indications from target data centers. The replicator component selectively communicates a data object to a target data center based on the selection indication of that target data center. The replicator component also produces profile data for a data object, the profile data being representative of content of the data object. The receiver component receives profile data from a source data center. The receiver component then sends a selection indication, determined based on the file selection criteria and the profile data, to the source data center.

Independent **claim 24** relates to a data system comprising plural data centers and plural client systems. Each data center includes a data storage component, a file server, a replicator component, and a collection of file selection criteria provided from other data centers. The replicator component produces profile data for a data object, the profile data being representative of content of the data object. The replicator component selectively communicates

a data object to a target data center based on the profile data and the selection criteria corresponding to that target data center.

Independent **claim 27** relates to a data system comprising plural data centers, each data center having plural client systems. The data system further comprises a selection server in communication with the data centers. Each data center includes a data storage component, a file server, and a replicator component. The replicator component produces profile data for a data object and communicates it to the selection server, the profile data being representative of content of the data object. The replicator component receives selection indicators from the selection server, wherein the data object is selectively communicated to target data centers based on the selection indicator. The selection server includes a collection of selection criteria received from the data centers, and produces the selection indicators based on the profile data and on the collection of selection criteria.

U.S. Patent No. 5,790,886 Allen

The patent to Allen discloses a method and system for automatically allocating space within a data storage system for multiple data sets which may include units of data, databases, files or objects. Each data set preferably includes a group of associated preference/requirement parameters which are arranged in a hierarchical order and then compared to corresponding data storage system characteristics for available devices. The data set preference/requirement parameters may include performance, size, availability, location, portability, share status and other attributes which affect data storage system selection. Data storage systems may include solid-state memory, disk drives, tape drives, and other peripheral storage systems. Data storage system characteristics may thus represent available space, cache, performance, portability, volatility, location, cost, fragmentation, and other characteristics which address user needs. The data set preference/requirement parameter hierarchy is established for each data set, listing each parameter from a "most important" parameter to a "least important" parameter. Each attempted storage of a data set will result in an analysis of all available data storage systems and the creation of a linked chain of available data storage systems representing an ordered sequence of preferred data storage systems. Data storage system selection is then

performed utilizing this preference chain, which includes all candidate storage systems. As illustrated, the user or the system may select a plurality of data set parameters which, as described above, may include performance, size, availability, location, portability, share status and other attributes which affect data storage system selection. (See, e.g., Abstract and column 6, lines 7-34).

As to **claim 1**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and producing profile information that is based on the content of a data object. The reference does not show or suggest the data object is selectively copied from its location in a first data storage system to a second data storage system based on the selection criteria and on the profile information.

As to **claim 12**, the reference does not show or suggest a data storage system having a data processing component that produces profile information comprising information associated with content of a data object. The reference does not show or suggest a data processing component that compares selection criteria with the profile information, where the selection criteria are associated with a second data server and determine whether the data object is copied to the second data server. The reference does not show or suggest a data processing component that performs the copying depending on the outcome of the comparison.

As to **claim 18**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and storing the selection criteria on a first data storage system. The reference does not show or suggest the data object is selectively copied from its location in the first data storage system to a second data storage system based on the selection criteria and on the profile information.

As to **claim 20**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component that receives selection indications from target data centers and selectively communicates a data object to a target data center based on the selection indication of that target data center. The reference does not show or suggest that the replicator component also produces profile data for a data object, where the profile data is representative of content of the data object. The reference does not show or suggest the data center including a receiver component

that receives profile data from a source data center, and which sends a selection indication, determined based on the file selection criteria and the profile data, to the source data center.

As to **claim 24**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component and a collection of file selection criteria. The reference does not show or suggest that the replicator component produces profile data for a data object, the profile data being representative of content of the data object and that it selectively communicates a data object to a target data center based on the profile data and the selection criteria corresponding to that target data center.

As to **claim 27**, the reference does not show or suggest a data system comprising plural data centers, where each data center includes a replicator component that produces profile data for a data object and communicates it to the selection server, the profile data being representative of content of the data object. The reference does not show or suggest that the replicator component receives selection indicators from the selection server, wherein the data object is selectively communicated to target data centers based on the selection indicator. The reference does not show or suggest that the selection server includes a collection of selection criteria received from the data centers, and produces the selection indicators based on the profile data and on the collection of selection criteria.

U.S. Patent No. 6,035,351 Billings et al.

The patent to Billings et al. discloses storage of user defined type file data in corresponding select physical format. Storing data on a data processing system is done upon generation of a data file by displaying a user interface allowing user selection of storage criteria for the data file. Responsive to user selection of storage criteria for a file determining a physical format type for the file from a plurality of available physical format types. Then the file is stored on a direct access storage device as at least a first record conforming with the determined physical format type. Implementation of the invention includes providing for user initiated editing and modification of the file descriptor to control physical aspects of storage of a file on auxiliary storage. Data may be written to a direct access storage device having a predefined

physical file format, in which case data is directed to the areas having the preferred format types for the data, or to a device where physical file format is selectable. The physical file format relates to the arrangement and data density of data tracks to which the data of a file is written. (See, e.g., Abstract and column 5, lines 18-35).

As to **claim 1**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and producing profile information that is based on the content of a data object. The reference does not show or suggest the data object is selectively copied from its location in a first data storage system to a second data storage system based on the selection criteria and on the profile information.

As to **claim 12**, the reference does not show or suggest a data storage system having a data processing component that produces profile information comprising information associated with content of a data object. The reference does not show or suggest a data processing component that compares selection criteria with the profile information, where the selection criteria are associated with a second data server and determine whether the data object is copied to the second data server. The reference does not show or suggest a data processing component that performs the copying depending on the outcome of the comparison.

As to **claim 18**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and storing the selection criteria on a first data storage system. The reference does not show or suggest the data object is selectively copied from its location in the first data storage system to a second data storage system based on the selection criteria <u>and</u> on the profile information.

As to **claim 20**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component that receives selection indications from target data centers and selectively communicates a data object to a target data center based on the selection indication of that target data center. The reference does not show or suggest that the replicator component also produces profile data for a data object, where the profile data is representative of content of the data object. The reference does not show or suggest the data center including a receiver component

that receives profile data from a source data center, and which sends a selection indication, determined based on the file selection criteria and the profile data, to the source data center.

As to **claim 24**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component and a collection of file selection criteria. The reference does not show or suggest that the replicator component produces profile data for a data object, the profile data being representative of content of the data object and that it selectively communicates a data object to a target data center based on the profile data and the selection criteria corresponding to that target data center.

As to **claim 27**, the reference does not show or suggest a data system comprising plural data centers, where each data center includes a replicator component that produces profile data for a data object and communicates it to the selection server, the profile data being representative of content of the data object. The reference does not show or suggest that the replicator component receives selection indicators from the selection server, wherein the data object is selectively communicated to target data centers based on the selection indicator. The reference does not show or suggest that the selection server includes a collection of selection criteria received from the data centers, and produces the selection indicators based on the profile data and on the collection of selection criteria.

U.S. Publication No. 2002/0065835 Fujisaki

The published patent application of Fujisaki discloses a file system assigning a specific attribute to a file, a file management method assigning a specific attribute to a file, and a storage medium on which is recorded a program for managing files. In a file system configured by one or a plurality of volumes, policy attribute data is set in correspondence with the path information of a directory, and a file is managed based on the policy attribute data. As a result, a policy specific to the directory can be set while maintaining the compatibility with an existing file system. For example, a volume number is set as the policy attribute data of a file, so that a file system administrator can specify the storage location of the file. First of all, attribute data to be processed, which is possessed by a parent directory, is obtained from metadata (information

for managing data such as the attribute, contents, storage location, etc. of data). The attribute data to be processed (policy attribute data), which is possessed by the registered policy data, is compared with the obtained. Then, it is determined whether or not the attribute data of the parent directory is inherited according to the inheritance attribute defined for each attribute data. If it is determined that the attribute data of the parent directory is inherited, this data is assigned to the target directory. If it is determined that the attribute data of the parent directory is not inherited, specified attribute data is assigned to the target directory. (See, e.g., Abstract and paragraph 95).

As to **claim 1**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and producing profile information that is based on the content of a data object. The reference does not show or suggest the data object is selectively copied from its location in a first data storage system to a second data storage system based on the selection criteria and on the profile information.

As to **claim 12**, the reference does not show or suggest a data storage system having a data processing component that produces profile information comprising information associated with content of a data object. The reference does not show or suggest a data processing component that compares selection criteria with the profile information, where the selection criteria are associated with a second data server and determine whether the data object is copied to the second data server. The reference does not show or suggest a data processing component that performs the copying depending on the outcome of the comparison.

As to **claim 18**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and storing the selection criteria on a first data storage system. The reference does not show or suggest the data object is selectively copied from its location in the first data storage system to a second data storage system based on the selection criteria <u>and</u> on the profile information.

As to **claim 20**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component that receives selection indications from target data centers and selectively communicates a data object to a target data center based on the selection indication of that target

data center. The reference does not show or suggest that the replicator component also produces profile data for a data object, where the profile data is representative of content of the data object. The reference does not show or suggest the data center including a receiver component that receives profile data from a source data center, and which sends a selection indication, determined based on the file selection criteria and the profile data, to the source data center.

As to **claim 24**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component and a collection of file selection criteria. The reference does not show or suggest that the replicator component produces profile data for a data object, the profile data being representative of content of the data object and that it selectively communicates a data object to a target data center based on the profile data and the selection criteria corresponding to that target data center.

As to **claim 27**, the reference does not show or suggest a data system comprising plural data centers, where each data center includes a replicator component that produces profile data for a data object and communicates it to the selection server, the profile data being representative of content of the data object. The reference does not show or suggest that the replicator component receives selection indicators from the selection server, wherein the data object is selectively communicated to target data centers based on the selection indicator. The reference does not show or suggest that the selection server includes a collection of selection criteria received from the data centers, and produces the selection indicators based on the profile data and on the collection of selection criteria.

U.S. Publication No. 2002/0147734 Shoup et al.

The published patent application of Shoup et al. discloses a policy based archiving system that receives data files in various formats and with various attributes. The archiving system examines each data file's attributes to correlate each data file with at least one policy by employing policy predicates. A policy is a collection of actions and decisions relating to the various storage and processing modules of the archiving system. In one aspect, the archiving system scans the content of a received data file to correlate the data file to a policy in

accordance with the semantic content of the data file. The data file attributes are examined in accordance with the policy predicates (step 84). In one embodiment, policy predicates dictate that the semantic content of the data file is examined to extract key terms and phrases. In this embodiment, the extracted content is compared to predefined content to correlate the data file to a policy in accordance with the data file's semantic content. In one embodiment, the data file's semantic content is parsed by employing a parsing algorithm. The parsing algorithm preferably searches for content in accordance with rules. (See, e.g., Abstract and paragraph 23).

As to **claim 1**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and producing profile information that is based on the content of a data object. The reference does not show or suggest the data object is selectively copied from its location in a first data storage system to a second data storage system based on the selection criteria and on the profile information.

As to **claim 12**, the reference does not show or suggest a data storage system having a data processing component that produces profile information comprising information associated with content of a data object. The reference does not show or suggest a data processing component that compares selection criteria with the profile information, where the selection criteria are associated with a second data server and determine whether the data object is copied to the second data server. The reference does not show or suggest a data processing component that performs the copying depending on the outcome of the comparison.

As to **claim 18**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and storing the selection criteria on a first data storage system. The reference does not show or suggest the data object is selectively copied from its location in the first data storage system to a second data storage system based on the selection criteria <u>and</u> on the profile information.

As to **claim 20**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component that receives selection indications from target data centers and selectively communicates a data object to a target data center based on the selection indication of that target data center. The reference does not show or suggest that the replicator component also produces

profile data for a data object, where the profile data is representative of content of the data object. The reference does not show or suggest the data center including a receiver component that receives profile data from a source data center, and which sends a selection indication, determined based on the file selection criteria and the profile data, to the source data center.

As to **claim 24**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component and a collection of file selection criteria. The reference does not show or suggest that the replicator component produces profile data for a data object, the profile data being representative of content of the data object and that it selectively communicates a data object to a target data center based on the profile data and the selection criteria corresponding to that target data center.

As to **claim 27**, the reference does not show or suggest a data system comprising plural data centers, where each data center includes a replicator component that produces profile data for a data object and communicates it to the selection server, the profile data being representative of content of the data object. The reference does not show or suggest that the replicator component receives selection indicators from the selection server, wherein the data object is selectively communicated to target data centers based on the selection indicator. The reference does not show or suggest that the selection server includes a collection of selection criteria received from the data centers, and produces the selection indicators based on the profile data and on the collection of selection criteria.

U.S. Publication No. 2002/0174306 Gajjar et al.

The published patent application of Gajjar et al. discloses a storage provisioning policy that is created by specifying storage heuristics for storage attributes using storage heuristic metadata. Storage attributes characterize a storage device and storage heuristic metadata describe how to specify a storage heuristic. Using the storage heuristic metadata, storage heuristics are defined to express a rule or constraint as a function of a storage attribute. In addition, the storage provisioning policy may also specify mapping rules for exporting the storage to a consumer of the storage, such as the server or server cluster. In an embodiment, a

method for creating one or more storage provisioning policies is provided. The method comprises: defining one or more storage attributes; defining one or more storage heuristic metadata associated with the one or more storage attributes; and specifying one or more storage heuristics using the defined one or more storage heuristic metadata associated with the one or more defined storage attributes to create the storage provisioning policy, the storage provisioning policy usable to provision a storage device, wherein the provisioned storage device includes discoverable data that satisfies the storage heuristics for the storage attributes. (See, e.g., Abstract and paragraphs 8-10).

As to **claim 1**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and producing profile information that is based on the content of a data object. The reference does not show or suggest the data object is selectively copied from its location in a first data storage system to a second data storage system based on the selection criteria and on the profile information.

As to **claim 12**, the reference does not show or suggest a data storage system having a data processing component that produces profile information comprising information associated with content of a data object. The reference does not show or suggest a data processing component that compares selection criteria with the profile information, where the selection criteria are associated with a second data server and determine whether the data object is copied to the second data server. The reference does not show or suggest a data processing component that performs the copying depending on the outcome of the comparison.

As to **claim 18**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and storing the selection criteria on a first data storage system. The reference does not show or suggest the data object is selectively copied from its location in the first data storage system to a second data storage system based on the selection criteria <u>and</u> on the profile information.

As to **claim 20**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component that receives selection indications from target data centers and selectively communicates a data object to a target data center based on the selection indication of that target

data center. The reference does not show or suggest that the replicator component also produces profile data for a data object, where the profile data is representative of content of the data object. The reference does not show or suggest the data center including a receiver component that receives profile data from a source data center, and which sends a selection indication, determined based on the file selection criteria and the profile data, to the source data center.

As to **claim 24**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component and a collection of file selection criteria. The reference does not show or suggest that the replicator component produces profile data for a data object, the profile data being representative of content of the data object and that it selectively communicates a data object to a target data center based on the profile data and the selection criteria corresponding to that target data center.

As to **claim 27**, the reference does not show or suggest a data system comprising plural data centers, where each data center includes a replicator component that produces profile data for a data object and communicates it to the selection server, the profile data being representative of content of the data object. The reference does not show or suggest that the replicator component receives selection indicators from the selection server, wherein the data object is selectively communicated to target data centers based on the selection indicator. The reference does not show or suggest that the selection server includes a collection of selection criteria received from the data centers, and produces the selection indicators based on the profile data and on the collection of selection criteria.

U.S. Publication No. 2003/0229637 Baxter et al.

The published patent application of Baxter et al. discloses a computer implemented method for safeguarding files, comprising the steps of designating a location on a first computer for storage of files to be safeguarded, selecting certain of the files to be safeguarded from the location based upon predetermined selection criteria, copying the selected files to be safeguarded to a second computer, deleting the selected files from the first computer, processing the selected files to be safeguarded on the second computer, and storing the selected

files to be safeguarded in a restricted access database. In a second embodiment, the file is copied to a second computer, but not deleted from the first computer, in addition to all other steps of the method. The invention also includes an apparatus for carrying out the methods of the invention. The system is capable of interpreting the content of a file to provide searchable text. (See, e.g., Abstract, paragraphs 131 and following, and claim 20).

As to **claim 1**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and producing profile information that is based on the content of a data object. The reference does not show or suggest the data object is selectively copied from its location in a first data storage system to a second data storage system based on the selection criteria and on the profile information.

As to **claim 12**, the reference does not show or suggest a data storage system having a data processing component that produces profile information comprising information associated with content of a data object. The reference does not show or suggest a data processing component that compares selection criteria with the profile information, where the selection criteria are associated with a second data server and determine whether the data object is copied to the second data server. The reference does not show or suggest a data processing component that performs the copying depending on the outcome of the comparison.

As to **claim 18**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and storing the selection criteria on a first data storage system. The reference does not show or suggest the data object is selectively copied from its location in the first data storage system to a second data storage system based on the selection criteria <u>and</u> on the profile information.

As to **claim 20**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component that receives selection indications from target data centers and selectively communicates a data object to a target data center based on the selection indication of that target data center. The reference does not show or suggest that the replicator component also produces profile data for a data object, where the profile data is representative of content of the data object. The reference does not show or suggest the data center including a receiver component

that receives profile data from a source data center, and which sends a selection indication, determined based on the file selection criteria and the profile data, to the source data center.

As to **claim 24**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component and a collection of file selection criteria. The reference does not show or suggest that the replicator component produces profile data for a data object, the profile data being representative of content of the data object and that it selectively communicates a data object to a target data center based on the profile data and the selection criteria corresponding to that target data center.

As to **claim 27**, the reference does not show or suggest a data system comprising plural data centers, where each data center includes a replicator component that produces profile data for a data object and communicates it to the selection server, the profile data being representative of content of the data object. The reference does not show or suggest that the replicator component receives selection indicators from the selection server, wherein the data object is selectively communicated to target data centers based on the selection indicator. The reference does not show or suggest that the selection server includes a collection of selection criteria received from the data centers, and produces the selection indicators based on the profile data and on the collection of selection criteria.

U.S. Publication No. 2004/0039891 Leung et al.

The published patent application of Leung et al. discloses optimizing storage capacity utilization based upon data storage costs. Techniques for optimizing capacity utilization among multiple storage units based upon costs associated with storing data on the storage units. Embodiments of the present invention automatically determine when data movement is needed to optimization storage utilization for a group of storage units. According to an embodiment of the present invention, in order to optimize storage utilization and storage cost, files are moved from a source storage unit to a target storage unit that has a lower data storage cost associated with it than the source storage unit. The storage units may be assigned to one or more servers. The "file selection criteria information" specifies information identifying conditions related to

files. According to an embodiment of the present invention, the selection criteria information for a placement rules specifies one or more clauses (or conditions) related to an attribute of a file such as file type, relevance score of file, file owner, etc. (See, e.g., Abstract and paragraph 116).

As to **claim 1**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and producing profile information that is based on the content of a data object. The reference does not show or suggest the data object is selectively copied from its location in a first data storage system to a second data storage system based on the selection criteria and on the profile information.

As to **claim 12**, the reference does not show or suggest a data storage system having a data processing component that produces profile information comprising information associated with content of a data object. The reference does not show or suggest a data processing component that compares selection criteria with the profile information, where the selection criteria are associated with a second data server and determine whether the data object is copied to the second data server. The reference does not show or suggest a data processing component that performs the copying depending on the outcome of the comparison.

As to **claim 18**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and storing the selection criteria on a first data storage system. The reference does not show or suggest the data object is selectively copied from its location in the first data storage system to a second data storage system based on the selection criteria and on the profile information.

As to **claim 20**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component that receives selection indications from target data centers and selectively communicates a data object to a target data center based on the selection indication of that target data center. The reference does not show or suggest that the replicator component also produces profile data for a data object, where the profile data is representative of content of the data object. The reference does not show or suggest the data center including a receiver component that receives profile data from a source data center, and which sends a selection indication, determined based on the file selection criteria and the profile data, to the source data center.

As to **claim 24**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component and a collection of file selection criteria. The reference does not show or suggest that the replicator component produces profile data for a data object, the profile data being representative of content of the data object and that it selectively communicates a data object to a target data center based on the profile data and the selection criteria corresponding to that target data center.

As to **claim 27**, the reference does not show or suggest a data system comprising plural data centers, where each data center includes a replicator component that produces profile data for a data object and communicates it to the selection server, the profile data being representative of content of the data object. The reference does not show or suggest that the replicator component receives selection indicators from the selection server, wherein the data object is selectively communicated to target data centers based on the selection indicator. The reference does not show or suggest that the selection server includes a collection of selection criteria received from the data centers, and produces the selection indicators based on the profile data and on the collection of selection criteria.

Foreign Publication No. EP 0617373 Burket et al.

The published patent application of Burket et al. discloses a distributed storage system in which an originating storage location establishes the criteria for storage management for a file. When the file is transmitted to other, subsidiary storage locations, it is accompanied by information controlling the storage of the file. For example, the duration of storage will be controlled by the control information. When a master file is deleted from an archive, in accordance with the criteria established at the time of storage, copies of the file at subsidiary locations can either be rendered inaccessible or alternately the storage management for that file can be changed. A feature is the distribution of storage management control information along with the file to diverse storage locations in a complex data processing system. In accordance with the invention, an image object distribution manager processor includes a memory having a management class table in which is stored a user-defined policy for managing the storage of

objects at diverse storage locations in a network. The management class table can specify document types and storage classes and for each document type and storage class, the table can provide for the user-defined period for retention of the document in both its master copy form and its derivative copy form. (See, e.g., Abstract and column 2, lines 26-39).

As to **claim 1**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and producing profile information that is based on the content of a data object. The reference does not show or suggest the data object is selectively copied from its location in a first data storage system to a second data storage system based on the selection criteria and on the profile information.

As to **claim 12**, the reference does not show or suggest a data storage system having a data processing component that produces profile information comprising information associated with content of a data object. The reference does not show or suggest a data processing component that compares selection criteria with the profile information, where the selection criteria are associated with a second data server and determine whether the data object is copied to the second data server. The reference does not show or suggest a data processing component that performs the copying depending on the outcome of the comparison.

As to **claim 18**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and storing the selection criteria on a first data storage system. The reference does not show or suggest the data object is selectively copied from its location in the first data storage system to a second data storage system based on the selection criteria <u>and</u> on the profile information.

As to **claim 20**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component that receives selection indications from target data centers and selectively communicates a data object to a target data center based on the selection indication of that target data center. The reference does not show or suggest that the replicator component also produces profile data for a data object, where the profile data is representative of content of the data object. The reference does not show or suggest the data center including a receiver component

that receives profile data from a source data center, and which sends a selection indication, determined based on the file selection criteria and the profile data, to the source data center.

As to **claim 24**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component and a collection of file selection criteria. The reference does not show or suggest that the replicator component produces profile data for a data object, the profile data being representative of content of the data object and that it selectively communicates a data object to a target data center based on the profile data and the selection criteria corresponding to that target data center.

As to **claim 27**, the reference does not show or suggest a data system comprising plural data centers, where each data center includes a replicator component that produces profile data for a data object and communicates it to the selection server, the profile data being representative of content of the data object. The reference does not show or suggest that the replicator component receives selection indicators from the selection server, wherein the data object is selectively communicated to target data centers based on the selection indicator. The reference does not show or suggest that the selection server includes a collection of selection criteria received from the data centers, and produces the selection indicators based on the profile data and on the collection of selection criteria.

Foreign Publication No. GB 2367163 Martin et al.

The published patent application of Martin et al. discloses optimized selection and accessing of stored files. A method, apparatus, and computer program are disclosed for a computer-implemented technique for generating file copies with minimal mounting and positioning of storage volumes. The method receives a request to generate file copies specifying file selection criteria, identifies matching files meeting the selection criteria (e.g., type of file, file name, etc.), locates the matching files on their storage volumes, and copies the files to a copy set. Determination of file copying order is optimized by placing greater emphasis on relative storage locations of matching files than on the order in which their copies are requested. The method ensures that each matching file is included, without duplication, in the copy set. The end result is

that files are selected based on filter criteria of the inventory view, but are transferred without excessive mounting or positioning of volumes, according to the storage view. (See, e.g., Abstract, p. 9, lines 14-21, p. 10, lines 10-14).

As to **claim 1**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and producing profile information that is based on the content of a data object. The reference does not show or suggest the data object is selectively copied from its location in a first data storage system to a second data storage system based on the selection criteria and on the profile information.

As to **claim 12**, the reference does not show or suggest a data storage system having a data processing component that produces profile information comprising information associated with content of a data object. The reference does not show or suggest a data processing component that compares selection criteria with the profile information, where the selection criteria are associated with a second data server and determine whether the data object is copied to the second data server. The reference does not show or suggest a data processing component that performs the copying depending on the outcome of the comparison.

As to **claim 18**, the reference does not show or suggest a method for distributing data among data storage systems that includes obtaining selection criteria and storing the selection criteria on a first data storage system. The reference does not show or suggest the data object is selectively copied from its location in the first data storage system to a second data storage system based on the selection criteria <u>and</u> on the profile information.

As to **claim 20**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component that receives selection indications from target data centers and selectively communicates a data object to a target data center based on the selection indication of that target data center. The reference does not show or suggest that the replicator component also produces profile data for a data object, where the profile data is representative of content of the data object. The reference does not show or suggest the data center including a receiver component that receives profile data from a source data center, and which sends a selection indication, determined based on the file selection criteria and the profile data, to the source data center.

As to **claim 24**, the reference does not show or suggest a data system comprising plural data centers and plural client systems, where each data center includes a replicator component and a collection of file selection criteria. The reference does not show or suggest that the replicator component produces profile data for a data object, the profile data being representative of content of the data object and that it selectively communicates a data object to a target data center based on the profile data and the selection criteria corresponding to that target data center.

As to **claim 27**, the reference does not show or suggest a data system comprising plural data centers, where each data center includes a replicator component that produces profile data for a data object and communicates it to the selection server, the profile data being representative of content of the data object. The reference does not show or suggest that the replicator component receives selection indicators from the selection server, wherein the data object is selectively communicated to target data centers based on the selection indicator. The reference does not show or suggest that the selection server includes a collection of selection criteria received from the data centers, and produces the selection indicators based on the profile data and on the collection of selection criteria.

Conclusion

In view of this comments presented in the instant petition and the claim amendments presented in the accompanying preliminary amendment, the Examiner is respectfully requested to issue a first Office Action at an early date.

Respectfully submitted,

George B. F. Yee
Reg. No. 37,478

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
Tel: 650-326-2400 / Fax: 415-576-0300
Attachments
GBFY
60546654 v1

EXHIBIT A

TABLE OF CONENTS

60548785 v1

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 617 373 A2**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number: **94103265.8**

(51) Int. Cl.^{5}: **G06F 15/40**

(22) Date of filing: **04.03.94**

(30) Priority: **24.03.93 US 36305**

(43) Date of publication of application: **28.09.94 Bulletin 94/39**

(84) Designated Contracting States: **DE FR GB**

(71) Applicant: **International Business Machines Corporation**
Old Orchard Road
Armonk, N.Y. 10504 (US)

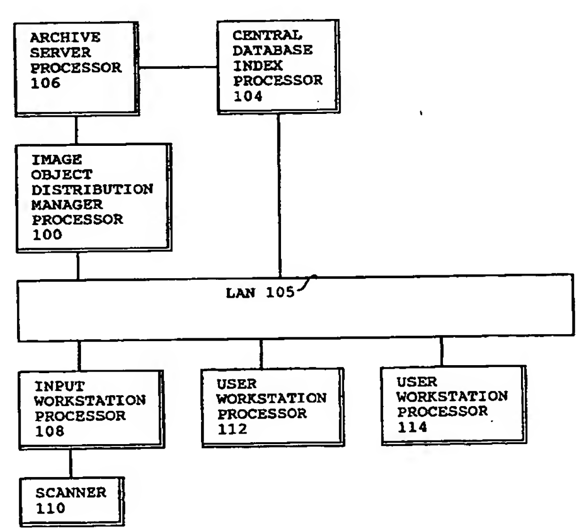
(72) Inventor: **Burket, Thomas G.**
9125 Wandering Trail Drive
Potomac, MD 20854 (US)
Inventor: **Daley, Michael E.**
7809 Polara Place
Rockville, MD 20855 (US)
Inventor: **Motter, Wesley J.**
23647 Rolling Fork Way
Gaithersburg, MD 20882 (US)
Inventor: **Tiwari, Manoj K.**
13336 Neerwinder Place
Germantown, MD 20874 (US)

(74) Representative: **Harrison, Robert John**
IBM Deutschland
Informationssysteme GmbH,
Patentwesen und Urheberrecht
D-70548 Stuttgart (DE)

(54) A method and system for parallel, system managed storage for objects on multiple servers.

(57) A distributed storage system is disclosed in which an originating storage location establishes the criteria for storage management for a file. When the file is transmitted to other, subsidiary storage locations, it is accompanied by information controlling the storage of the file. For example, the duration of storage will be controlled by the control information. When a master file is deleted from an archive, in accordance with the criteria established at the time of storage, copies of the file at subsidiary locations can either be rendered inaccessible or alternately the storage management for that file can be changed. A feature is the distribution of storage management control information along with the file to diverse storage locations in a complex data processing system.

FIGURE 1A



EP 0 617 373 A2

Background of the Invention

1. Technical Field

The invention disclosed broadly relates to data processing systems and more particularly relates to the management of storage in a data processing system.

2. Related Patents

The following patents are related to the invention disclosed herein and are incorporated herein by reference:
USP 5,093,911 by C. A. Parks, et al. entitled "Storage and Retrieval System," assigned to the IBM Corporation and incorporated herein by reference.

USP 5,161,214 by M. R. Addink, et al. entitled "Method and Apparatus for Document Image Management in a Case Processing System," assigned to the IBM Corporation and incorporated herein by reference.

3. Background Art

Systems Managed Storage (SMS) is the management of data objects in a storage hierarchy within a single system. Objects move through DASD, optical shelf libraries, tape, and so on within the confines of the single system. In an environment with distributed servers, a single SMS process for objects is insufficient. The problem is as follows:
Objects captured and stored at one server location may require movement to another server for permanent archival storage or for temporary use at that server.

When an object moves to another server for archival purposes, it may need to assume different management characteristics.

When an object is copied to another server for temporary use, it may need to assume different management characteristics.

When an object has been moved to another server for archival purposes and a copy has been left behind in the original server, the copy may need to assume different management characteristics.

Current SMS solutions do not cover these problems because of their limitation to single servers.

Objects of the Invention

It is therefore an object of the invention to provide an improved method for managing storage in a data processing system.

It is still another object of the invention to provide an improved method for parallel, system managed storage for objects on multiple servers in a data processing system.

Summary of the Invention

These and other objects, features and advantages are accomplished by the invention disclosed herein. A distributed storage system is disclosed in which an originating storage location establishes the criteria for storage management for a file. When the file is transmitted to other, subsidiary storage locations, it is accompanied by information controlling the storage of the file. For example, the duration of storage will be controlled by the control information. When a master file is deleted from an archive, in accordance with the criteria established at the time of storage, copies of the file at subsidiary locations can either be rendered inaccessible or alternately the storage management for that file can be changed. A feature is the distribution of storage management control information along with the file to diverse storage locations in a complex data processing system.

In accordance with the invention, an image object distribution manager processor includes a memory having a management class table in which is stored a user-defined policy for managing the storage of objects at diverse storage locations in a network. The management class table can specify document types and storage classes and for each document type and storage class, the table can provide for the user-defined period for retention of the document in both its master copy form and its derivative copy form. The table can also provide for the period of existence of the master file in the system before it is automatically archived in an archive server in the network.

Further in accordance with the invention, when an object file is transferred from one storage location to another storage location in the network, a data set of storage management attributes is transferred to the destination storage location. The destination storage attributes are defined either by the management class table in the image object distribution manager's processor, or alternately by local storage policy values for some of those attributes, which can be defined by the user for each respective storage device in the network.

In this manner, the overall system policy for the management of object storage in the network can be implemented throughout the system, and yet the user can define local customized options for that management policy to be applied at local storage devices in the network.

Brief Description of the Drawings

These and other objects, features and advantages of the invention will be more fully appreciated with reference to the accompanying figures.

Fig. 1A is an architectural block diagram of a first example data processing system within which the invention finds application.

Fig. 1B is an architectural block diagram of a second example data processing system within which the invention finds application.

Fig. 1C is a more detailed block diagram of the image object distribution manager (IODM) processor 100.

Fig. 1D is a more detailed block diagram of the archive server processor 106.

Fig. 1E is a more detailed block diagram of the input workstation processor 108.

Fig. 1F is a more detailed block diagram of the user workstation processor 112.

Fig. 1G is a more detailed block diagram of the user workstation processor 114.

Fig. 2 is a flow diagram of the sequence of operational steps depicting the migration between an original site and archival site.

Fig. 3 is a flow diagram of a sequence of operational steps which depicts the migration of file control in a more detailed manner.

Fig. 4 is a flow diagram of a sequence of operational steps depicting the process for the receipt of file control information by a target server.

Fig. 5 is a flow diagram of a sequence of operational steps for the retrieval by a local server from a remote server, of file control information.

Fig. 6A illustrates an example application of the invention, for the date of 3 March 1993, inputting the Jones document.

Fig. 6B continues the example of Fig. 6A, for the date of 4 March 1993, sending the master file to the user workstation 112 and keeping a copy at the input workstation 108.

Fig. 6C continues the illustration of Fig. 6B for the date of 6 March 1993, archiving the master file at the archive 106, keeping a copy of the file at the user workstation 112, and deleting a copy of the file at the input workstation 108.

Fig. 6D continues the illustration of Fig. 6C, for the date of 7 March 1993, sending a copy of the file to the user workstation 114.

Fig. 6E modifies the example of Fig. 6B by returning to the date of 5 March 1993, and sending the master file to the user workstation 114 and keeping a copy at the user workstation 112.

Fig. 6F continues the example of Fig. 6E at a later time on the date of 5 March 1993, archiving the master file to the archive 106 and keeping a copy at the user workstation 114. Discussion of the Preferred Embodiment

In accordance with the invention, when an object is captured into a local server, it is assigned attributes that define its Systems Managed Storage (SMS) characteristics and policies, such as how long to stay at certain points in the storage hierarchy. Examples: how long to reside on DASD before migration to optical storage, how long to reside on optical before migration to tape, and so on. The invention extends the characteristics and policies as follows:

* Define a new policy that controls how long the new object resides locally before being sent to an archival site.
* The new policy includes rules that allow the object to be given SMS attributes for its archival site by the original capture site or by the archival site, or both, with the actions of the archival site taking precedence. The archival site then manages the object in parallel with the management of the original site.
* The new policy includes rules that allow the object to be deleted when it has been archived at another site.
* The new policy includes rules that allow the object to be given new SMS attributes at the local site after it has been archived at another site. This allows an object to be managed differently after it has made a transition from being the original master copy of an object to being a copy of the now-archived original.
* When an object is deleted at the archival site, either by manual action or as a consequence of the management policies (e.g., automatic expiration), copies of the object stored at other sites may no longer be referenced.
* The archival process may occur at any time during the lifetime of the local object (the lifetime runs from object capture to expiration). The archival process may occur at any point in the storage hierarchy.
* When a remote site requests a copy of an object stored locally, the remote site receives the SMS attributes of the object as well as the object itself, and the remote site may

retain those attributes or assign its own attributes and policies independent of the original.

* Object copies and object originals may have identical or different storage management policies, depending on the local rules. For example, copies may have a limited lifetime or copies might be limited to higher-speed storage media intended for temporary, quick access. Copies may also be placed on permanent archival media as desired within the policies.
* The execution of the policies, once established, is automatic and does not require explicit manual action.

The architectural diagrams of Fig. 1A and Fig. 1B, provide examples of network of storage devices, with an input site, an archive site and one or more user sites, in a distributed data processing system.

Token ring local area network 105 is an example of a system of servers which can input, store, transfer and display document image objects. Each server can include a workstation processor with a hard disk drive storage, such as an IBM PS/2 Mod 80, for example. Input workstation server 108 is connected to LAN 105, and has a document scanner 110 as the input device. Workstation server 112 is connected to LAN 105. Workstation server 114 is connected to LAN 105. The archive server 106 is connected to LAN 105 in Fig. 1B, and has an optical mass storage device for archiving document images. A central data base server 104 is connected to LAN 105, for providing an index to all documents stored in the system. The image object distribution manager (IODM) server 100 connected to the LAN 105, stores a master set of control tables which define the storage policy for the various types of documents stored in the system and the various classes of storage in the system. The archive server 106 may be connected through the IODM server 100 to the LAN 105 in an alternate embodiment shown in Fig. 1A.

Figs. 1A and 1B show a first workstation 108 which includes a first storage which can receive a storage object, for example the image of a document which has been scanned into the system. In accordance with the invention, the storage object at workstation 108 can be stored in accordance with a storage management policy by means of storing control characters in association with the object, in the first workstation 108. For example, the duration of storage of the storage object may be encoded in the control characters which are stored in association with the storage of the object at workstation 108.

Figs. 1A and 1B also show a second workstation 112 which includes a second storage capable of storing objects such as image objects. In accordance with the invention, when an application or user desires to transfer a copy of the stored object at workstation 108 to the second storage device in the workstation 112, the storage management control characters stored in association with the stored object are transferred along with the stored object from the first workstation 108 to the second workstation 112. There, in workstation 112, both the copy of the object to be stored and the associated storage management characters are stored.

Further in accordance with the invention, a local clock can be associated with or contained in the first workstation 108, for the purpose of timing processes carried out at workstation 108. In accordance with the invention, the control characters associated with the object stored in the first in workstation 108, can specify a duration of storage for the object or alternately an instant in time for the initiation of further action at workstation 108 on the stored object. Such further action can include the deletion of the stored object, the transmission of the stored object to a second device, such as the workstation 112, or other storage management function.

Still further in accordance with the invention, a local clock can be associated or contained in the second workstation 112, to time processes carried out in the second workstation. In accordance with the invention, the control characters associated with the copy of the object stored in the second workstation 112, can specify the duration of storage for the copy of the object stored in workstation 112. Still further, the control characters can specify the instant in time at which other storage management functions can be performed on the copy of the object stored in workstation 112, through a monitoring of the local clock in the workstation 112 and a comparison of the local time with the timing information stored or contained in the control characters stored in association with the copy of the object in workstation 112.

Still further in accordance with the invention, a universal expiration instant can be specified in the control characters associated with the original copy of the object stored in workstation 108 and the derivative copy of the object stored in workstation 112. When the universal instant of time occurs, corresponding to a system-wide instant specified by a system-wide clock, the original copy of the object at workstation 108 and the derivative copy of the object in workstation 112 can be simultaneously or substantially simultaneously deleted from the respective storage devices in the two workstations.

Figs. 1A and 1B also show a central data base index 104 which stores an inverted file index that relates object files stored in the system, to their

respective storage location addresses. In accordance with the invention, if the control characters associated with an object stored at a storage location in the system, such as the first storage device in the workstation 108, specifies the transfer of a copy of the object to another storage device or specifies the deletion of the object from a storage device such as the workstation 108, then a message is transmitted from the processor associated with the storage device, such as the workstation 108, to the central data base index 104, to update the inverted file index information to reflect the change in the location or the change in existence of the stored object file.

Figs. 1A and 1B also show an archive server 106 which is connected to LAN 105 and which may also be connected to the central data base index 104. In accordance with the invention, the derivative copy of the object stored at the second storage device in the second workstation 112 in association with its control characters for that stored object, can have its storage control specified by its control characters as follows. After a specified duration or at a specified instant in time, a copy of the derivative copy, that is a third copy of the object, can be transferred from the second workstation 112 to the archive server 106 for archive storage in the archive server 106.

Figs. 1A and 1B also show an additional user workstation 114 which can be coupled to the token ring LAN, for accessing copies of the derivative copy of the object stored in the workstation 112 in response to application or user requests.

Thus it is seen that storage management control can be distributed along with an object file to be stored, by associating with the object, file control information in the form of control characters which are transmitted in associated with the object file to diverse storage locations in a complex data processing system.

Fig. 1C illustrates the image object distribution manager (IODM) processor 100. Processor 100 includes the memory 120 connected by means of the bus 125 to the CPU 122, the keyboard and display 124, and the LAN adapter 126 which couples the processor 100 to the LAN 105. Also included is the host adapter 127 connecting the bus 125 to the archive server processor 106 in the embodiment shown in Fig. 1A. Also shown in Fig. 1C is the object storage 128 connected to the bus 125, which can be a combination of magnetic hard disk drives and/or optical storage in both the read only and the read/write or the write-once-read-many embodiments.

The memory 120 stores the management class table 121 which embodies the user-defined system managed storage policy for the data processing system of Fig. 1A or Fig. 1B. The management class table 121 is accessed by the document type and the storage class. Document types can be correspondence or they can be buckslips or other types of document images intended to be stored in storage devices located throughout the network of Fig. 1A or Fig. 1B. The management class table 121 provides for a document type of "correspondence" or "CORR" which provides, for example, with a retention period measured in days for both master copies and derivative copies of a document. The retention period can be customized for storage classes of input workstations which may be different from storage classes of user workstations which may be different from storage classes of archive servers in the network of Fig. 1A or Fig. 1B. The management class table 121 also provides for the period during which a master document may reside on input workstations and user workstations in the network prior to its being automatically archived in an archive server processor, for example the archive server processor 106 of Figs. 1A or 1B.

The memory 120 of Fig. 1C also includes the image object distribution management program 123 which is shown in greater detail in the pseudo code of Table 1. The programs stored in the memory 120 are sequences of executable instructions which are executed by the CPU 122. The memory 120 of Fig. 1C can also store a data base searching program 118 which will provide searching support for searching the management class table 121 when input query of document type, storage class, and master or copy status of a particular document is input to the processor 100. The memory 120 of Fig. 1C will also include a document image management program 116, such as has been described in the related patent applications by C. A. Parks, et al. and by M. R. Addink, et al., cited above. The memory 120 of Fig. 1C will also include an operating system program 115. A better understanding of the operation of the IODM processor 100 can be gained by referring to the example shown in Figs. 6A-6F, as described below.

Fig. 1D is a detailed block diagram of the archive server processor 106. The processor 106 includes the memory 130 which is connected by means of the bus 135 to the CPU 132, the keyboard and display 134, and the LAN adapter 136 which connects to the LAN 105. Also included is the index adapter 137 which connects to the central data base index processor 104. Also included is the object storage 138, which can be combinations of magnetic hard disk drives and/or optical storage devices of the read only, read/write or write-once-read many types.

The memory 130 includes the local storage policy value attributes 182 which provide user-defined local options for a storage policy to be applied to the archive server processor 106. The

memory 130 also includes the archive memory catalog 131 which provides a partition for the storage of control attributes which are supplied by the IODM processor 100 and specifically from the management class table 121 therein, to implement a system-wide storage management policy.

The memory 130 also includes a local policy override partition 183, which the user or system administrator can use to indicate whether the local storage policy values 182 or the system-wide storage policy stored in the archive memory catalog 131 is to be applied in storing a particular file or class of files. Also included in the memory 130 of Fig. 1D is the image object distribution manager program 133 shown in the pseudo code of Table 4. The programs stored in the memory 130 are sequences of executable instructions which are executed by the CPU 132.

Also included in the archive server processor 106 is the object access method program 139 which facilitates the accessing of objects stored in the object storage 138. Also included in memory 130 is the document image management program 116 and the operating system program 115. A better understanding of the operation of the archive server processor 106 can be had by reference to the example shown in Figs. 6A-6F, which will be discussed below.

Fig. 1E is a detailed block diagram of the input workstation processor 108. Processor 108 includes the memory 140 which is connected by means of the bus 145 to the CPU 142, the keyboard and display 144, and the LAN adapter 146 which connects to the LAN 105. Also included is the scanner adapter 147 which connects to the scanner 110 and the object storage 148. The object storage 148 can be a combination of magnetic hard drives and optical storage devices of the read only, read/write or write-once-read-many type.

Also included in the input workstation processor 108 is the local storage policy attribute values 180 in the memory 140, which stores user-defined values for some of the control attributes for storing particular classes of files in the processor 108. Also included in the memory 140 is the input memory catalog 141 which is a partition for storing the storage control attributes received from the IODM processor 100 and in particular from the management class table 121 therein, to implement a system-wide storage policy.

The memory 140 also includes a local policy override partition 181, which stores a user provided yes or no value to indicate whether the local storage policy attribute values 180 are to override the system-wide storage policy values provided in the input memory catalog 141. The memory 140 also includes the image object distribution manager program 143 which is shown in pseudo code in Table 2. The programs in the memory 140 are sequences of executable instructions which are executed by the CPU 142. Also included in the memory 140 is the intelligent forms processing program 149 which can be used to perform a character recognition of information on document images scanned in by the scanner 110, to provide the automatic generation of alphanumeric control strings which would substitute for the control strings input by the keyboard 144. Examples of such control information would be document type, file name or other information typically required when a new document is scanned into the system. The intelligent forms processing program 149 is described in greater detail in the copending patent application serial number 07/870,129 by T. S. Betts, et al. entitled "Data Processing System and Method for Sequentially Repairing Character Recognition Errors for Scanned Images of Document Forms," filed April 15, 1992, assigned to the IBM Corporation and incorporated herein by reference.

Also included in the memory 140 is the document image management program 116 and operating system program 115. A better understanding of the operation of the input workstation processor 108 can be had by referring to Figs. 6A-6F describing an example, which will be discussed below.

Fig. 1F illustrates the user workstation processor 112. Processor 112 includes the memory 150 which is connected by means of the bus 155 to the CPU 152, the keyboard and display 154 and the LAN adapter 156 which connects to the LAN 105. Also included is an I/O adapter 157 enabling the processor 112 to communicate with other networks. Also included is the object storage 158 which can be a magnetic hard drive storage combined with an optical storage of the read only, read/write or write-once-read-many type.

The memory 150 also includes the local storage policy attribute values 184 which are user-defined storage policy values for particular types of documents to be stored in the processor 112. The memory 150 also includes the user memory catalog 151 which is a partition for storing storage policy attributes received from the management class table 121 of the IODM processor 100. The memory 150 also includes the local policy override partition 185 which enables the user to decide whether to apply the local storage policy attribute values 184 or alternately the system-wide storage policy values in the user memory catalog 151.

The memory 150 also includes the image object distribution manager program 153 which is shown in greater detail of the pseudo code of Table 3. The programs stored in the memory 150 are sequences of executable instructions which are executed in the CPU 152. Also included in the memory 150 is the document image management

program 116 and the operating system program 115. A better appreciation for the operation of the user workstation processor 112 can be had by reference to the example illustrated in Figs. 6A-6F, which will be discussed below.

Fig. 6G is a detailed block diagram of the user workstation processor 114. The user workstation processor 114 includes the memory 160 which is connected by means of the bus 165 to the CPU 162, the keyboard and display 164 and the LAN adapter 166 which is connected to the LAN 105. Also included is the I/O adapter 167 which allows the processor 114 to be connected to other networks. Also included is the object storage 168 which can be a magnetic hard drive storage combined with an optical storage of the read only, read/write, or write-once-read-many type.

The memory 160 also includes the local storage policy attribute values 186 which are attribute values defined by the user for specific application at the processor 114, to implement a local policy for specific types of documents stored at processor 114. Also included in the memory 160 is the user memory catalog 161 which is a partition for storing storage attribute values received from the IODM processor and in particular from the management class table 121 therein, which embody the system-wide storage management policy.

Also included in the memory 160 is the local policy override partition 187 which stores an indication provided by the user as to whether the local storage policy attribute values 186 or alternately the system-wide storage policy values in user memory catalog 161 are to be applied. Also included in the memory 160 is the image object distribution manager program 163 which is shown in pseudo code in Table 3. The programs stored in the memory 160 are sequences of executable instructions which are executed by the CPU 162. Also included in the memory 160 is the document image management program 116 and the operating system program 115. A better understanding of the operation of the user workstation processor 114 can be had by referring to the example illustrated in Figs. 6A-6F, which will be discussed below.

The memory catalog partitions shown for the input workstation 108, the user workstation 112, the archive server 106 and the user workstation 114, are all shown with a similar format. The input memory catalog 141, the user memory catalog 151, the archive catalog 131 and the user memory catalog 161 all include an update partition for storing the last date upon which that partition was written into. Each of these memory catalogs also includes a store class 170 which stores the storage class for that particular processor. Each memory catalog also includes a file name partition 171 which stores the file name for a corresponding file which is stored in the object storage for that processor. The memory catalog also includes a master/copy partition 172, which stores the status of the file as being either a master copy or a derivative copy which is currently stored in the object storage of that processor. Each memory catalog also includes a document type partition 173 which stores the document type, for example "correspondence" or "CORR" as a first document type or "buckslips" as a second document type, for example. Each memory catalog also includes a data made partition 174 which stores the date upon which the master document image was scanned into the system by the scanner 110. The memory catalogs also include the archive period partition 175 which provides the period during which a master document image may reside in the system following the date made 174, before the master file is transferred to the archive storage 106. Each memory catalog also includes the archive date partition 176 which is the date upon which the master file is to be archived. The memory catalogs also include the retention period partition 177 which stores the period during which the file may reside in the object storage of the processor, starting from the day it was initially stored in the object storage. The memory catalogs are include the retention date partition which stores the date upon which the file is removed from the object storage of the processor. The partitions illustrated in Figs. 1D-1F and in Figs. 6A-6F are merely illustrative of the kinds of attributes which can be provided in the memory catalog for these classes of storage devices and processors. Not all of the memory catalogs for all of the processors in the network of Fig. 1A or Fig. 1B need be identical. Fewer or additional storage control attributes can be provided by the memory catalogs in each of these processors. An understanding of the operation and use of the memory catalogs can be had by referring to the example illustrated in Figs. 6A-6F which will be discussed below.

Fig. 2 illustrates a flow diagram of the sequence of operational steps for carrying out the process of migration of file control information between an original site and an archival site. The following are the steps shown in Fig. 2.

Steps and Their Discussion for Fig. 2

202. Original hard copy source document is submitted to the input device, such as scanned in by a document scanner 110 at workstation 108.

204. Document image is received from the input device 110 and stored temporarily on the hard disk drive at the workstation 108.

206. In some scenarios, a user may choose to review the new document at that time and assign attributes to the document, such as keywords, account number, or priority of action. This step may be augmented by intelligent document processing, as described in the copending patent application serial number 07/870,129 by T. S. Betts, et al. entitled "Data Processing System and Method for Sequentially Repairing Character Recognition Errors for Scanned Images of Document Forms," filed April 15, 1992, assigned to the IBM Corporation and incorporated herein by reference.
208. Either by explicit user action or by internal system processing, the document is assigned a set of management policies.
210. System software stores the document content on the target server 112 and enters storage control information about the document in the server's catalog.
212. System software examines the document information in the catalog and determines how many days should pass before the document is migrated to an archival site 106. Server's 112 catalog is set to indicate the date at which migration is scheduled.
214. On or after the date scheduled for this document, system software examines the server's 112 catalog for documents that require action. System software determines this document requires action - migration to an archive site 106.
216. See flow "Migration Detail" of Fig. 3, which completes.
218. Because the original document content is being moved to the archival site 106, the server 112 marks its remaining copy of the document as a "copy" distinguished from "master" or "original," by altering the storage control information in the server's 112 catalog.
220. Server 112 looks at the management policies for the document. One policy establishes what to do to documents that have been migrated to another server, e.g. archive 106. That is, delete or preserve the document on server 112.
222. If the policy indicates deletion, then the server 112 deletes its copy of the document from its storage space and catalog. Flow completes.
224. If the policy indicates no deletion (preserve copy), then the document is not deleted from server 112. Check other management policies for the document. A policy may allow the attributes of a document to be changed after it has been migrated, because a document copy may be managed differently than its master.
226. If the local policy for this type of document does not allow attributes to change after migration, leave attributes alone. Only the "copy" vs. "master" indication has changed. Go to step 230.
228. If the local policy for this type of document allows attributes to change, then system software or customized user code determines new attributes for the document.
230. Compute next management action for the document based on its attributes and the current date.

Fig. 3 is a flow diagram of a sequence of operational steps which provides a more detailed illustration of the migration of file control characters. The migration detail flow steps are as follows:

Migration Detail Flow of Fig. 3

302. Local server policy identifies which server in the network is the target archival server 106 for this server's 112 captured documents.
304. If local server 112 policy does not allow a document to be given new policy attributes as it migrates, then skip to step 308.
306. System software or customized user code determines the attributes to be sent with the object to its new server 106. These attributes do not affect the attributes in the local server's 112 catalog.
308. Send document content and attributes to new server 106. The attributes sent are either the active attributes. Then, go to flow "Receipt by Target Server" in Fig. 4.
312. Target server 106 notifies source server 112 of successful receipt.
314. Source server 112 updates central catalog in IODM 100 for all servers to record that the document "master" is now on the target server.

Fig. 4 is a flow diagram of a sequence of operational steps which depicts the process for receipt of file control information by a target server. The following are the steps for the receipt by the target server.

Receipt by Target Server for Fig. 4

402. Target server 106 receives document and the suggested attributes from the source server 112.
404. If target server's 106 local policy does not allow attributes to be sent from another server, skip to step 408.
406. Validate that attributes sent by source server 112 are acceptable. Override attributes that are not acceptable, using system software or customized user code. Skip to step 410.

408. Use target server's 106 local policy for migrated documents to assign policy attributes to the received document.

410. Store document in target server's 106 storage space and in its catalog.

412. Use document's policy attributes and their meaning at the target server 106 to schedule next action for the document at that server. The action might include migrating the document to a third server for more permanent archival, deleting it, or other storage management action. This action is independent of document management at the original source server 108 that captured the document. Step 414 determines if the target server 106 policy allows attributes for time, based on current date and management policies for that server and document.

Fig. 5 is a flow diagram of a sequence of operational steps for performing the process of retrieval of file control information by a local server 114 from a remote server 106. The following steps describe the retrieval from a remote server.

Retrieval From Remote Server for Fig. 5

502. Local server 114 determines it needs document that it does not have, and determines which remote server 106 has the document master, using the control index 104.

504. Local server 114 requests copy of the document from the server 106 with the master.

506. Owning server 106 sends document and relevant policy attributes to the requesting local server 114.

508. If local server's 114 policy does not allow acceptance of attributes sent from another server, skip to 512.

510. Validate that attributes sent by remote server 106 are acceptable. Override attributes that are not acceptable, using system software or customized user code. Skip to step 514.

512. Use local server's 114 policy for new copies of documents to assign policy attributes to the received document.

514. Store document and attributes in local server's 114 storage space and in its catalog.

516. Use document's policy attributes and their meaning at the target server 114 to schedule next action for the document at that server 114. The action may not include migrating the document to another server for more permanent archival, because this is not the master copy of the document, and only master copies may be migrated. This action is independent of document management at the owning remote server 106, and independent of whether this local server 114 was the same server that captured the original source document before migration.

EXAMPLE 1

A letter arrives to a business from a customer asking for the addition of a new person to an insurance policy. The business employee who handles incoming mail scans (110) the letter into the storage system (108) and, based on the customer number from the letter, puts it into the customer's folder. Because the letter is a policy change letter, the employee assigns the letter to management category POLICY__LETTER.

You could consider POLICY__LETTER a "management class," which is one of the set of "control characters." Now, previously the system storage administrator had given the management class the following other characteristics for managing this type of data. These control values were previously stored in IODM 100, and are now downloaded to server 108.

"Migration period" - 0 days. Migrate this object to the archive at the first opportunity. The customer wishes to have a permanent archive made the same day as images arrive into the system.

Storage class" - store the image on DASD. Class transition period" - 7 days. Keep the local image on DASD for one week. This is to keep a local copy handy during the likely period of processing by the company and potential follow-up with customer.

"Retention period" - 7 days. Delete the object at this site after one week. Instead of a class transition to a slower medium, the object will be purged.

The flows of Figs. 2-5 show changing attributes during the migration to the archive site 106. Here is an example of what could happen. In this example, the local site (112) decides to change the attributes when it sends the image to the archive site (106). It says the image's management class should be "ARCHIVED" which the system storage administrator previously defined in IODM 100. These values for "ARCHIVED" management class are now downloaded to server 106. It could have characteristics (at the target archive site 106):

"Migration period" - none. This class does not migrate anywhere because it is the final destination. "Storage class" - store the image on DASD.

"Class transition period" - 1 day. After one day, the image will be migrated to another storage class (optical) at this site. Note that this occurs independently of the seven-day period still running down at the original capture site. Retention period" - 7 years. For this company, this type of image must be retained for seven years.

EXAMPLE 2

A particular example of system managed storage in accordance with the invention, is provided in Figs. 6A-6F. The pseudo code represented in Tables 1-4 should also be referred to in discussing the example of Figs. 6A-6F. The example refers to the system of Figs. 1A or 1B and provides for the input of a hard copy Jones document into the input workstation 108. This is followed by the transfer of the master copy to the user workstation 112 and the retention of a derivative copy in the input workstation 108. This is followed by the transfer of the master copy to the archive server 106 and the retention of a derivative copy in the user workstation 112 and further, the expiration of the retention period for the copy in the input workstation 108. This is followed by the transfer of a derivative copy of the file to the user workstation 114. This sequence is illustrates in Figs. 6A-6D.

In Fig. 6A, the scanner 110 scans in the hard copy Jones document and the image file for that document is transferred to the input object storage 148 of the input workstation 108 where it is stored as a master copy. Substantially simultaneously, the document type input is provided either by the keyboard 144 for the input workstation 108, or alternately the document type is derived from character recognition on the scanned image from the scanner 110. The document type information is then transferred from the input processor 108 to the IODM processor 100. The IODM processor 100 will then use the document type input, which in this case is "correspondence" and the storage class for the input workstation 108, which is "input workstation," to select a retention period and an archive master period. Since it is a master copy which is to be stored in the input object storage 148, it will be the retention period for a master copy at an input workstation for a document type which is "correspondence" which is accessed from the management class table 121. Also, the archive master period will be selected for the input workstation storage class. The value of the retention period of five days and of the archive period three days is then output from the IODM processor 100 to the input workstation 108, where it is stored in the input memory catalog 141.

The following values are stored in the input memory catalog 141 at this time. First the update value of 3 March 1993 is stored. The storage class 170 is "input." The file name 171 is "Jones." The master/copy 172 is "master." The document type 173 is "CORR." The date made 174 is today's date of 3 March 1993. The archive period 175 is the value transferred from the IODM processor or three days. The archive date 176 is computed by adding the archive period 175 to the date made 174, which becomes 6 March 1993. The retention period 177 is the value transferred from the IODM processor 100 or five days. A retention date 178 is computed from the archive date 176 plus the retention period 177, which becomes 8 March 1993. These control attributes stored in the input memory catalog 141 of the input workstation 108, will provide the system-wide storage management policy for the storage of the Jones master file in the object storage 148. If the user has specified that the local policy override 181 in the input workstation processor 108 would be effective, then the local storage policy attribute values 180 would have been substituted for the corresponding values in the input memory catalog 141.

For the example shown in Figs 6A-6F, the local policy override partitions will all be set to "no" so that the system-wide storage policy will be implemented. Reference should be made to the pseudo code of Tables 1 and 2 for a detailed understanding of the sequence of steps carried out in storing the master file of the Jones document in the input object storage 148 and in entering the storage control attributes into the input memory catalog 141.

Fig. 6B illustrates the operation on the following day for March 1993 sending the master file from the input workstation 108 to the user workstation 112 and of keeping a copy of the file at the input workstation 108. As is shown in Fig. 6B, the storage class 170 for the user memory catalog 151 is "user." The file name 171 of "Jones" is transferred from the input workstation 108. The master/copy 172 is "master," the document type 173 is "CORR," the data made 174 is still the original scanning date of 3 March 1993. The archive period 175 is received from the management class table 121 of the IODM processor 100 and is a value of two days. The archive date 176 is computed by adding the archive period of two days to the date made 174, which becomes the archive date 176 of 5 March 1993. The retention period 177 is received from the management class table 121 of the IODM processor 100 and is a value of 10 days. This value is added to the update value of 4 March 1993, which is the date that the entry is made to the user memory catalog 151. Thus the retention date 178 is the retention period 177 of 10 days added to the update of 4 March 1993, or 14 March 1993. Reference should be made to the pseudo code of Tables 1, 2 and 3 to understand the detailed steps carried out in Fig. 6B.

Fig. 6C shows the action on 6 March 199, of archiving the master at the archive 106, keeping a copy at the user workstation 112 and deleting a copy at the input workstation 108 because it has expired. Reference should be made to Tables 1-4 for the pseudo code detailed representation of the

steps for Fig. 6C.

Fig. 6D shows the action on 7 March 1993 sending a copy to the user workstation 114. Reference should be made to the pseudo code in Tables 1-4 for a detailed description of the steps which accomplish Fig. 6D. Fig. 6E and Fig. 6F modify the example, taking the time back to 4 March 1993 in Fig. 6B. Following Fig. 6B, Fig. 6E then occurs on 5 March 1993, and sends the master to the user workstation 114 and keeps a copy at the user workstation 112. Reference should be made to the pseudo code of Tables 1-4 for detailed steps carried out in Fig. 6E.

Fig. 6F occurs at a later time on the same day of 5 March 1993, where the master is archived on the archive server 106 and a copy is kept at the user workstation 114 (see Tables 1-4 for details).

The transition of the example from Fig. 6B to Fig. 6C archives the master file at the archive 106 by transferring it from the user 112. A copy is to be kept at the workstation 112. Tables 1-4 illustrate the detailed steps for transferring from the management class table 121 in the IODM processor 100 the appropriate control attributes for system-wide storage policy to the user memory catalog 151 of the user workstation 112 and to the archive memory catalog 131 in the archive server 106. In particular, it can be seen that the archive period 175 and retention period 177 are transferred to the user memory catalog 151 from the management class table 121, and that the resulting archive date 176 and the retention date 178 are computed and written into the user memory catalog 151. Also it can be seen that the archive period 175 and the retention period 177 have been transferred to the archive memory catalog 131 for the archive server 106. From this, the archive date 176 and the retention date 178 are computed and written into the archive memory catalog 131. In this example, the local policy override for the user workstation 112 and the archive server 106 are set to "no" so that the system-wide policy established by the management class table 121 in the IODM processor 100, is applied at the user workstation 112 and the archive server 106.

In addition, it can be seen in Fig. 6C, that the retention date of 6 March 1993 for the copy of the file stored in the input workstation 108, as represented by the input memory catalog 141 of Fig. 6B, has expired on the date 6 March 1993, as represented in Fig. 6C. This results in the retention routine executed in the input workstation 108, deleting the copy of the file stored in the input object server 148 and deleting the entry for the Jones copy in the input memory catalog 141, as is shown in Fig. 6. This operation is set forth in greater detail in Table 2.

Thus it can be seen that the system-wide storage policy established by the system administrator and embodied in the management class table 121 in the IODM memory 120 of the IODM processor 100, is carried out in various servers and storage devices in the network.

Fig. 6D illustrates the transition in the example from Fig. 6C, for the following day of 7 March 1993, where a copy of the file stored in the archive server 106, is transferred to the workstation 114. Fig. 6D shows that the appropriate storage control attributes are output from the management class table 121 in the IODM processor 100 to the user memory catalog 161 in the user workstation 114, in association with the transfer from the archive server 106 of a copy of the Jones file to the object storage 168 in the user workstation 114. In particular, the archive period 175 and the retention period 177 are transferred from the management class table 121 to the user memory catalog 161 as shown in Fig. 6D. Then, the archive date 176 is computed and the retention date 178 is computed and those values are stored in the user memory catalog 161 of Fig. 6D. These steps are provided in the pseudo code of Table 3.

It can be seen that a retention date has been computed of 9 March 1993 for the copy stored at the user workstation 114. This carries out the system-wide storage policy established by the system administrator and embodied in the management class table 121. Reference can be made to Fig. 1G illustrating the user workstation processor 114, to see the effect of a local policy override. If the local policy override 187 indicates "yes," then the local storage policy values 186 will be substituted for system-wide policy values which have been stored in the user memory catalog 161 for the user workstation processor 114. In the case of a local policy override, the retention period 177 of 15 days, as provided in the local storage policy values 186 of Fig. 1G, will be substituted for the system-wide specified retention period 177 of two days which was provided by the IODM processor 100. This would result in a computed retention date 178 of the addition of 15 days to the 7 March 1993 update value, which would result in a 22 March 1993 retention date 178 being the effective retention date for the copy of the Jones file stored at the user workstation 114 on 7 March 1993. In this manner, a system-wide storage management policy can be given effect, and yet local user selection of alternate storage policies can be effectively substituted.

Regarding the transition from Fig. 6B on 4 March 1993 to Fig. 6C on 6 March 1993, it is noted that the archive date 176 in the user memory catalog 151 of Fig. 6B specifies 5 March 1993. However, as is observed in Fig. 6C, the date of

archiving is 6 March 1993. An example of this circumstance would be if the date 5 March 1993 fell on a holiday so that the user workstation 112 was not turned on on that day. As can be seen in the pseudo code of Table 3, when the user workstation is turned on, an initial inquiry is made as to whether there are any archive dates which are the same day or older than the current day. Thus, the master file stored in the user object storage 158 is archived on the day 6 March 1993 following the specified archive date 176 of 5 March 1993 for the user memorycatalog 151 of Fig. 6B.

To illustrate the feature of transferring a file between the same class of storage devices, Figs. 6E and 6F are provided for the example. Fig. 6E starts from Fig. 6B whose date is 4 March 1993, and on the following day 5 March 1993, Fig. 6E provides for the transfer of the master copy of the Jones file from the user workstation 112 to the user workstation 114. (This transfer before archiving would be possible in the example, if the user workstation 112 sent the file to workstation 114 after midnight, so that the date becomes 050393, but the device has remained on. Thus step 802 of Table 3 is not satisfied.) This example in Fig. 6E is a transfer of a file without changing its master/copy status and without changing its storage class status which remains "user." Since the document type 173 also does not change, no change in the storage control attributes are necessary in order to maintain the effect of the system-wide policy established in the management class table 121 of the IODM processor 100. Therefore, as can be seen in Fig. 6E, in the transfer of the master file from the user object store 158 in user workstation 112 to the user object store 168 in the user workstation 114, the contents of the user memory catalog 151 at the user workstation 112 is transferred to the user memory catalog 161 in the user workstation 114. At the user workstation 114, the extra copy which is received has the retention period 177 used to recompute the retention date 178, by adding the retention period of 10 days to the user memory catalog update of 5 March 1993, resulting in a new computed retention date 178 of 15 March 1993, which is written into the user memory catalog 161. In this manner, the storage control attributes provided for the storage of the master file in the user workstation 112, are transferred to the user workstation 114, and those attributes which must be recomputed such as the retention date 178, are recomputed and stored at the new location. In this manner, the system-wide storage management policy is given effect.

The transition from Fig. 6E to Fig. 6F occurs later on the same day of 5 March 1993, where the pseudo code of Table 3 determines that the master file stored in the user workstation 114 is ready to be archived in the archive server 106. The transfer of the master file from the user workstation 114 to the archive server 106 and the keeping of a copy of that file at the user workstation 114 is accompanied by the transfer of the appropriate storage control attributes from the management class table 121 in the IODM processor 100 to the user memory catalog 161 and the archive memory catalog 131, as shown in Fig. 6F. In particular, the archive period 175 and the retention period 177 are transferred into the user memory catalog 161 and the corresponding archive date 176 and the retention 178 are re-computed into the user memory catalog 161. The file name 171, master copy status 172, document type 173 and date made 174 are transferred directly from the user memory catalog 161 to the archive memory catalog 131. The archive period 175 and the retention period 177 are transferred to the archive memory catalog 131 from the management class table 121 of the IODM processor 100. The archive date 176 and the retention date 178 are then re-computed and stored in the archive memory catalog 131. In this manner, the system-wide storage management policy embodiment in the IODM processor 100, is given effect in the user workstation 114 and the archive server 106.

The invention can be applied to image management systems and to management of arbitrary machine-readable object types, since any object type is a potential candidate for storage management policies.

The resulting invention enables an originating storage location to establish the criteria for storage management for file throughout a complex, distributed storage data processing system.

Although a specific embodiment of the invention has been disclosed, it will be understood by those having skill in the art that changes can be made to that specific embodiment without departing from the spirit and the scope of the invention.

Claims

1. In a data processing system having a plurality of storage devices, including an originating storage device and an archiving storage device, a method for parallel system managed storage for objects, comprising the steps of:
receiving an object file at an originating storage location;
assembling control information to control the storage of said file in a storage device;
transmitting a copy of said file and said control information to a second storage device in said data processing system;
performing said control functions at said second storage device in response to said control

information.

2. The method of claim 1 which further comprises:
said control information controlling a duration of storage for said object file.

3. The method of claim 2 which further comprises:
deleting said object file at said originating storage location;
reading said control information at said second storage location associated with said copy of said file;
deleting said copy of said file at said second storage location, in response to said control information.

4. The method of claim 1, which further comprises:
selectively performing alternate local option control functions at said second storage device.

5. The method of claim 1, wherein said control information controls the migration of said object file to a third storage device in said system.

6. The method of claim 1, wherein said control information controls the migration of said object file to a third storage device in said system after a duration from said receiving at said originating location.

7. In a data processing system having a plurality of storage devices, including an originating storage device and an archiving storage device, a subsystem for parallel system managed storage for objects, comprising:
receiving means for receiving an object file at an originating storage location;
assembling means coupled to said receiving means, for assembling control characters to control the storage of said file in a storage device;
transmitting means coupled to said assembling means, for transmitting over a communication link a copy of said file and said control characters to a second storage device in said data processing system;
control means coupled to said communication link, for performing said control functions at said second storage device in response to said control characters.

8. The subsystem of claim 7 which further comprises:
said control characters controlling a duration of storage for said object file.

9. The subsystem of claim 8 which further comprises:
first deleting means coupled to said receiving means, for deleting said object file at said originating storage location;
reading means coupled to said control means, for reading said control characters at said second storage location associated with said copy of said file;
second deleting means coupled to said reading means, for deleting said copy of said file at said second storage location, in response to said control information.

10. The subsystem of claim 7, wherein said object file is an image object file.

11. The subsystem of claim 7, wherein said control information controls the migration of said object file to a third storage device in said system.

12. The subsystem of claim 7, wherein said control information controls the migration of said object file to a third storage device in said system after a duration.

FIGURE 1A

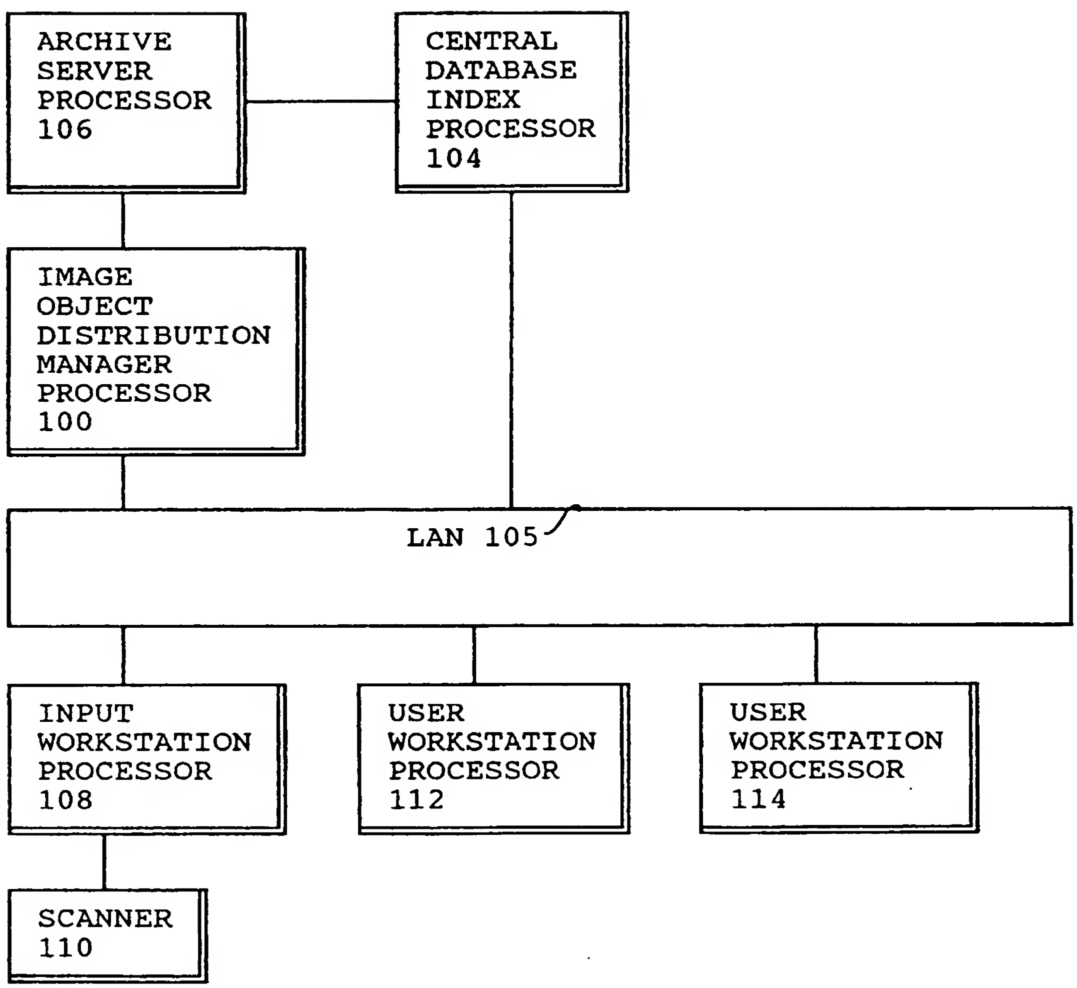

ARCHIVE
SERVER
PROCESSOR
106
CENTRAL
DATABASE
INDEX
PROCESSOR
104
IMAGE
OBJECT
DISTRIBUTION
MANAGER
PROCESSOR
100
LAN 105
INPUT
WORKSTATION
PROCESSOR
108
USER
WORKSTATION
PROCESSOR
112
USER
WORKSTATION
PROCESSOR
114
SCANNER
110

FIGURE 1B

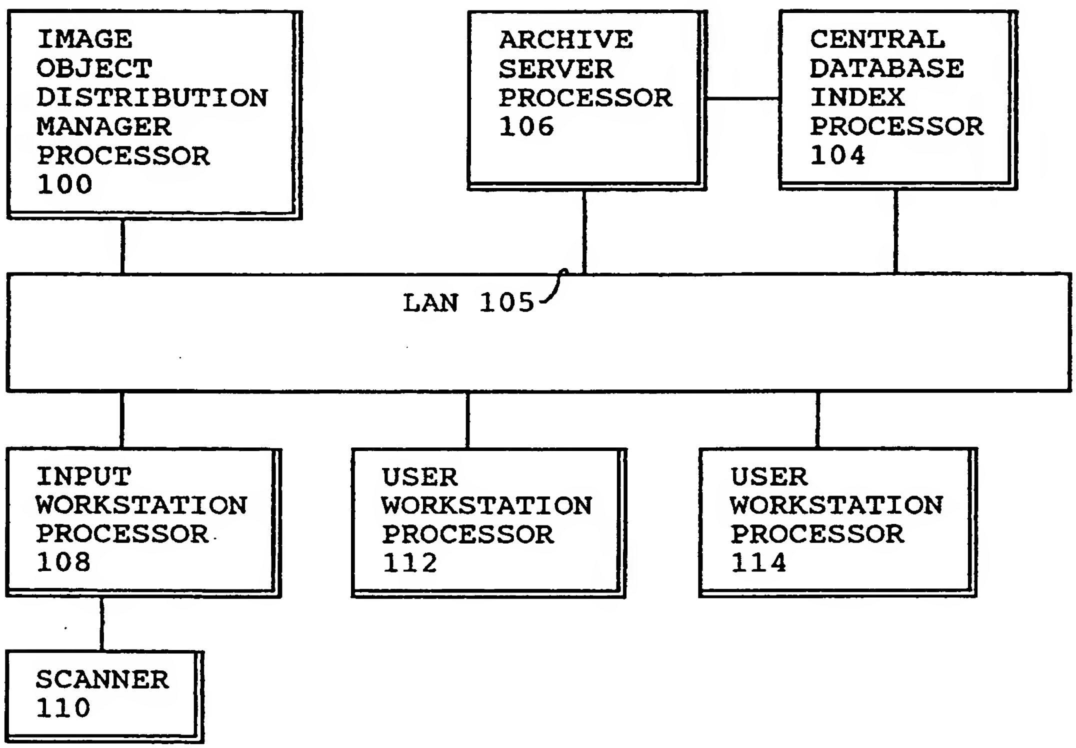
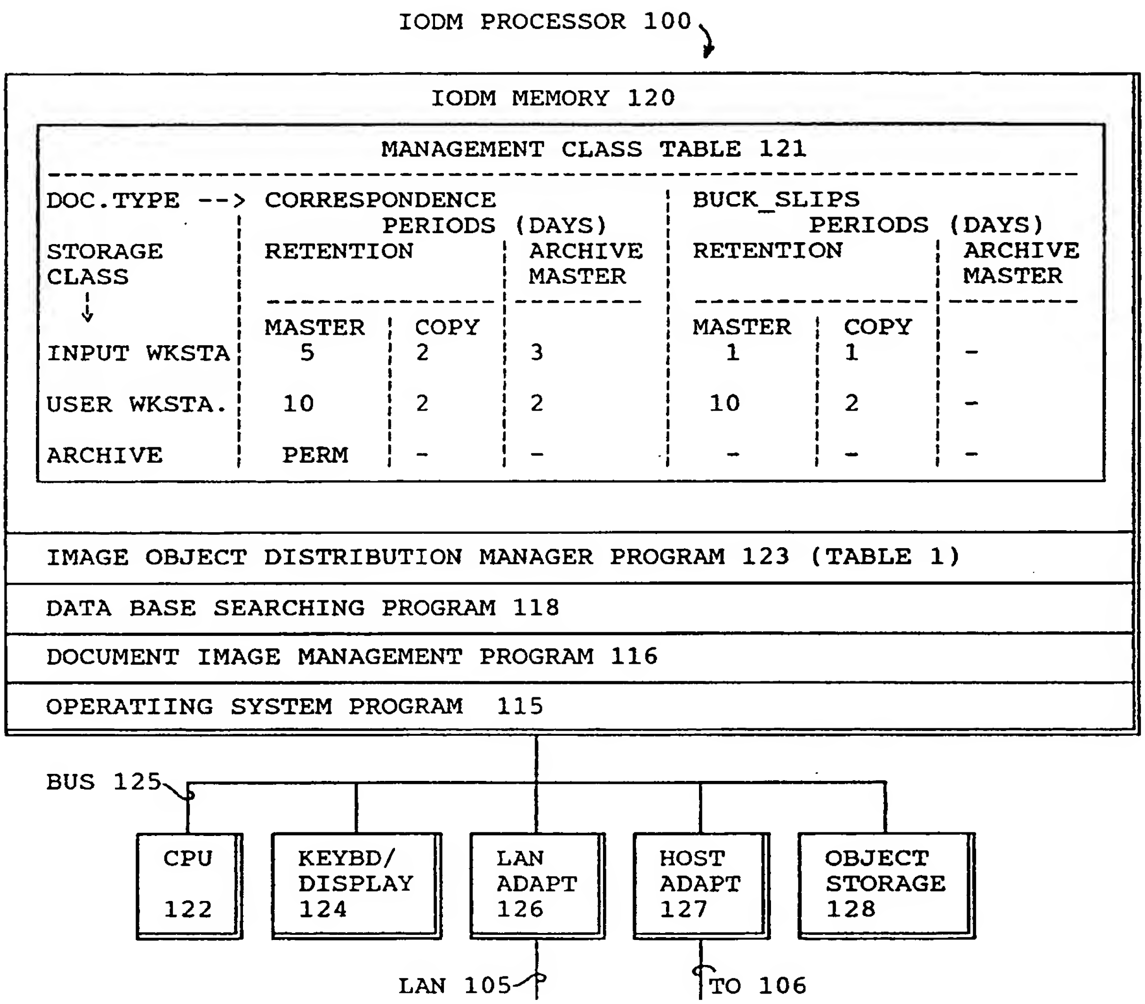


IMAGE OBJECT DISTRIBUTION MANAGER PROCESSOR 100
ARCHIVE SERVER PROCESSOR 106
CENTRAL DATABASE INDEX PROCESSOR 104
LAN 105
INPUT WORKSTATION PROCESSOR 108
USER WORKSTATION PROCESSOR 112
USER WORKSTATION PROCESSOR 114
SCANNER 110

FIGURE 1C



DOC.TYPE -->	CORRESPONDENCE PERIODS (DAYS)			BUCK_SLIPS PERIODS (DAYS)		
STORAGE CLASS ↓	RETENTION		ARCHIVE MASTER	RETENTION		ARCHIVE MASTER
	MASTER	COPY		MASTER	COPY	
INPUT WKSTA	5	2	3	1	1	-
USER WKSTA.	10	2	2	10	2	-
ARCHIVE	PERM	-	-	-	-	-

FIGURE 1 D

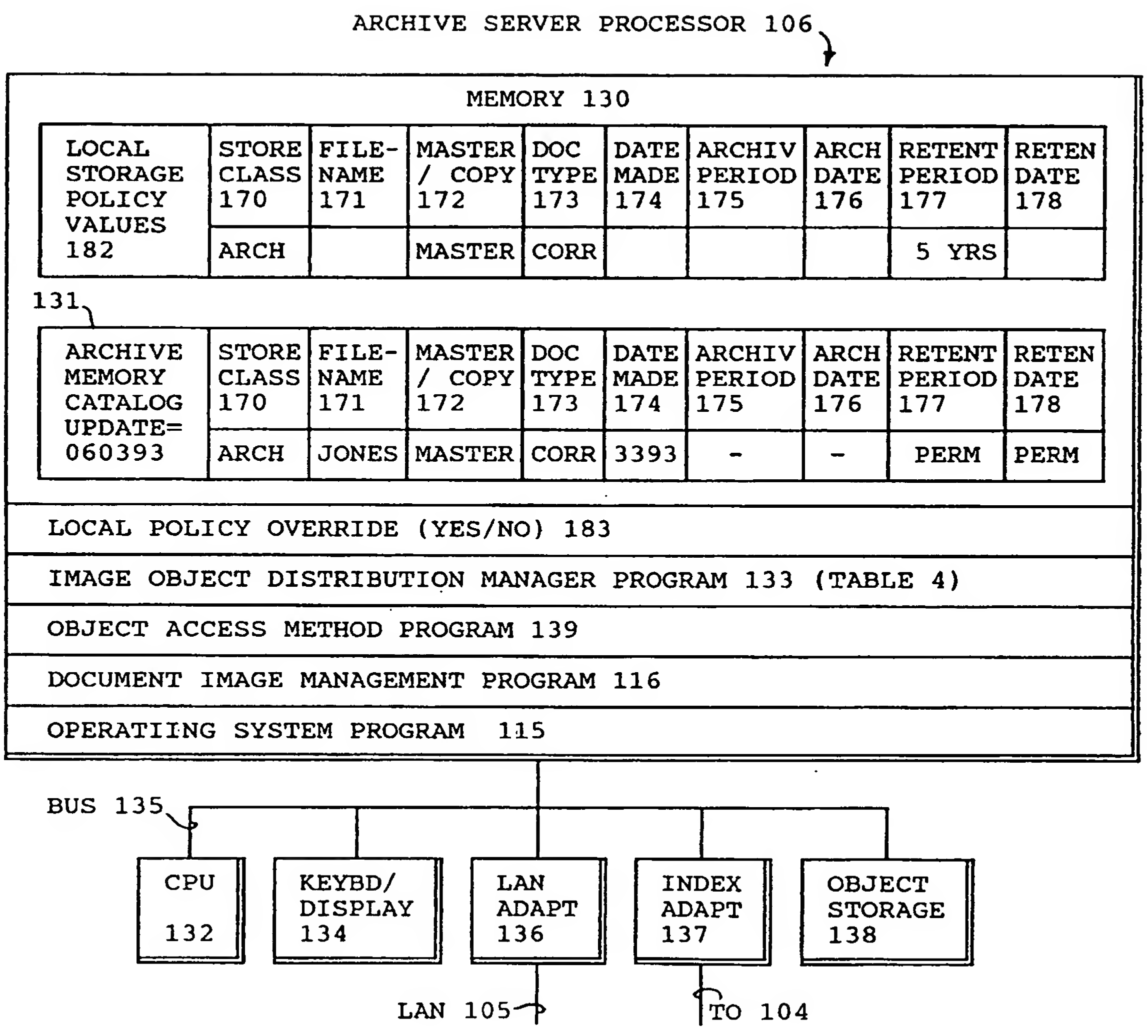

ARCHIVE SERVER PROCESSOR 106
MEMORY 130
LOCAL STORAGE POLICY VALUES 182
STORE CLASS 170
FILE-NAME 171
MASTER / COPY 172
DOC TYPE 173
DATE MADE 174
ARCHIV PERIOD 175
ARCH DATE 176
RETENT PERIOD 177
RETEN DATE 178
ARCH
MASTER
CORR
5 YRS
131
ARCHIVE MEMORY CATALOG UPDATE= 060393
STORE CLASS 170
FILE-NAME 171
MASTER / COPY 172
DOC TYPE 173
DATE MADE 174
ARCHIV PERIOD 175
ARCH DATE 176
RETENT PERIOD 177
RETEN DATE 178
ARCH
JONES
MASTER
CORR
3393
-
-
PERM
PERM
LOCAL POLICY OVERRIDE (YES/NO) 183
IMAGE OBJECT DISTRIBUTION MANAGER PROGRAM 133 (TABLE 4)
OBJECT ACCESS METHOD PROGRAM 139
DOCUMENT IMAGE MANAGEMENT PROGRAM 116
OPERATIING SYSTEM PROGRAM 115
BUS 135
CPU 132
KEYBD/ DISPLAY 134
LAN ADAPT 136
INDEX ADAPT 137
OBJECT STORAGE 138
LAN 105
TO 104

FIGURE 1E

INPUT WORKSTATION PROCESSOR 108

MEMORY 140

LOCAL STORAGE POLICY VALUES 180	STORE CLASS 170	FILE-NAME 171	MASTER / COPY 172	DOC TYPE 173	DATE MADE 174	ARCHIV PERIOD 175	ARCH DATE 176	RETENT PERIOD 177	RETEN DATE 178
	INPUT		MASTER	CORR		7		20	

141

INPUT MEMORY CATALOG UPDATE= 030393	STORE CLASS 170	FILE-NAME 171	MASTER / COPY 172	DOC TYPE 173	DATE MADE 174	ARCHIV PERIOD 175	ARCH DATE 176	RETENT PERIOD 177	RETEN DATE 178
	INPUT	JONES	MASTER	CORR	3393	3	6393	5	08393

LOCAL POLICY OVERRIDE (YES/NO) 181

IMAGE OBJECT DISTRIBUTION MANAGER PROGRAM 143 (TABLE 2)

INTELLIGENT FORMS PROCESSING PROGRAM 149

DOCUMENT IMAGE MANAGEMENT PROGRAM 116

OPERATIING SYSTEM PROGRAM 115

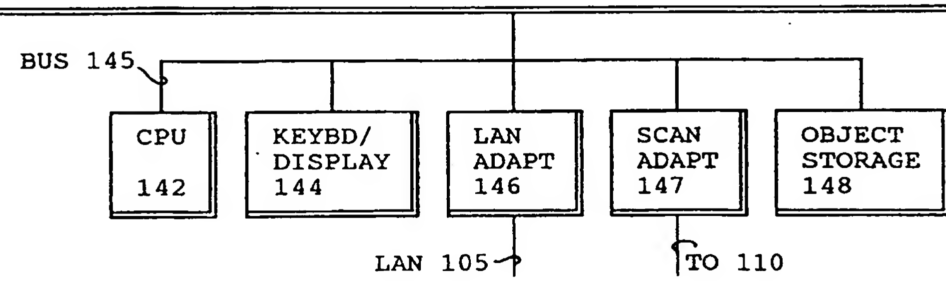


FIGURE 1F

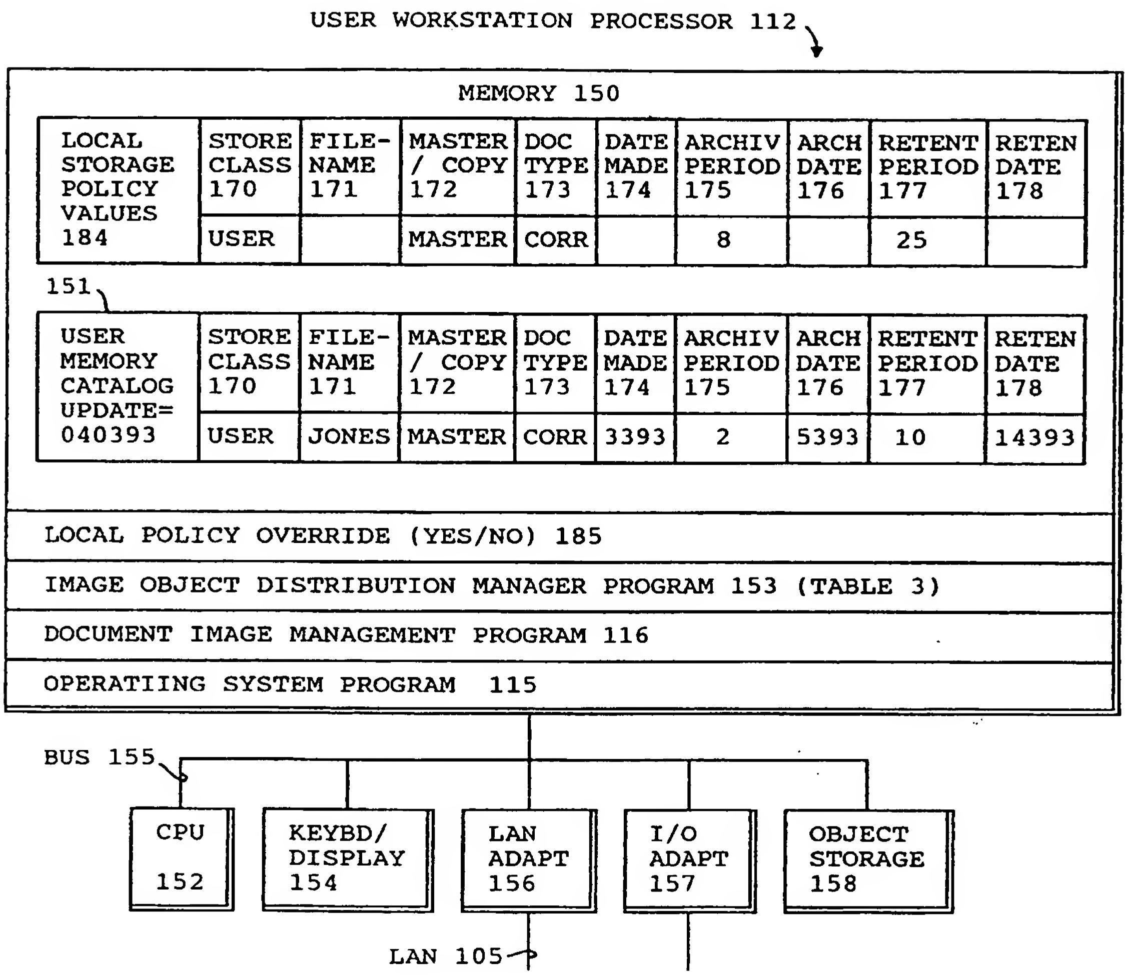

USER WORKSTATION PROCESSOR 112
MEMORY 150
LOCAL STORAGE POLICY VALUES 184
STORE CLASS 170
FILE-NAME 171
MASTER / COPY 172
DOC TYPE 173
DATE MADE 174
ARCHIV PERIOD 175
ARCH DATE 176
RETENT PERIOD 177
RETEN DATE 178
USER
MASTER
CORR
8
25
151
USER MEMORY CATALOG UPDATE= 040393
STORE CLASS 170
FILE-NAME 171
MASTER / COPY 172
DOC TYPE 173
DATE MADE 174
ARCHIV PERIOD 175
ARCH DATE 176
RETENT PERIOD 177
RETEN DATE 178
USER
JONES
MASTER
CORR
3393
2
5393
10
14393
LOCAL POLICY OVERRIDE (YES/NO) 185
IMAGE OBJECT DISTRIBUTION MANAGER PROGRAM 153 (TABLE 3)
DOCUMENT IMAGE MANAGEMENT PROGRAM 116
OPERATIING SYSTEM PROGRAM 115
BUS 155
CPU 152
KEYBD/ DISPLAY 154
LAN ADAPT 156
I/O ADAPT 157
OBJECT STORAGE 158
LAN 105

FIGURE 1G

USER WORKSTATION PROCESSOR 114

MEMORY 160

LOCAL STORAGE POLICY VALUES 186	STORE CLASS 170	FILE-NAME 171	MASTER / COPY 172	DOC TYPE 173	DATE MADE 174	ARCHIV PERIOD 175	ARCH DATE 176	RETENT PERIOD 177	RETEN DATE 178
	USER		COPY	CORR		10		15	

161

USER MEMORY CATALOG UPDATE= 070393	STORE CLASS 170	FILE-NAME 171	MASTER / COPY 172	DOC TYPE 173	DATE MADE 174	ARCHIV PERIOD 175	ARCH DATE 176	RETENT PERIOD 177	RETEN DATE 178
	USER	JONES	COPY	CORR	3393	-	-	2	09393

LOCAL POLICY OVERRIDE (YES/NO) 187

IMAGE OBJECT DISTRIBUTION MANAGER PROGRAM 163 (TABLE 3)

DOCUMENT IMAGE MANAGEMENT PROGRAM 116

OPERATIING SYSTEM PROGRAM 115

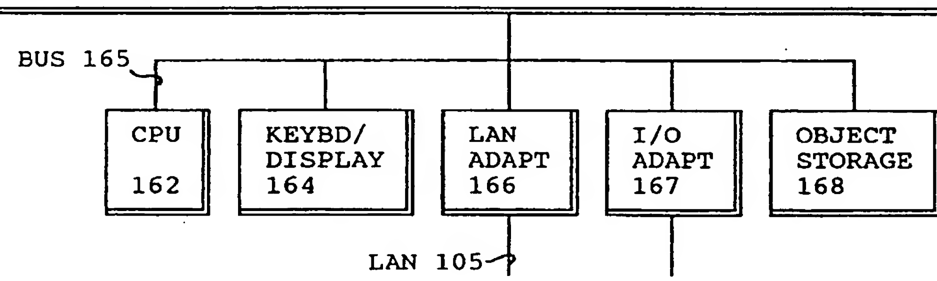


FIG. 2

MIGRATION BETWEEN
ORIGINAL SITE
AND ARCHIVAL SITE

DOCUMENT

202

INPUT DOCUMENT OBJECT

RECEIVE AT USER WORKSTATION 204

USER IDENTIFY'S DOCUMENT INDEX ATTRIBUTES 206

ATTRIBUTES DETERMINE DOCUMENT MANAGEMENT POLICIES 208
(POLICY INCLUDES HOW LONG A NEWLY CAPTURED
DOCUMENT STAYS ON THE LOCAL SERVER BEFORE
MIGRATION TO ARCHIVAL SITE)

DOCUMENT IS STORED ON SERVER AND ITS 210
POLICY ATTRIBUTES ARE ALSO STORED ON SERVER

MIGRATION TO ARCHIVAL SITE IS SCHEDULED FOR FUTURE 212
TIME BASED ON CURRENT DATE AND MANAGEMENT POLICIES
FOR THAT SERVER AND DOCUMENT

SERVER DETECTS THAT MIGRATION DATE HAS ARRIVED 214
AND BEGINS MIGRATION PROCESS (SEE OTHER FLOW)

MIGRATION TO ARCHIVAL SITE COMPLETES 216

DOCUMENT IS MARKED AS A "COPY" INSTEAD OF ORIGINAL 218
AT THE CAPTURE SITE (ORIGINAL IS NOW AT ARCHIVAL SITE)

DOES LOCAL POLICY SAY TO DELETE MIGRATED DOCUMENTS? 220

YES

222 DELETE DOCUMENT AND ITS ATTRIBUTES
FROM LOCAL SERVER

NO

DONE

DOES LOCAL POLICY ALLOW CHANGING ATTRIBUTES 224
OF LOCAL COPIES ONCE MIGRATED TO ARCHIVAL SITE?

YES

228 UPDATE POLICY ATTRIBUTES FOR THIS
DOCUMENT AT LOCAL SERVER

NO,
DO NOTHING 226

COMPUTE NEW MANAGEMENT ACTION DATE FOR THE DOCUMENT 230

FIG. 3

MIGRATION DETAIL FLOW

```
    SOURCE SERVER SELECTS ITS TARGET SERVER          302
                   |
                   V
      DOES SOURCE SERVER'S POLICY DEFINITION ALLOW THE
    SENDING OF ATTRIBUTES DIFFERENT THAN THE CURRENT
        VALUES FOR A MIGRATING DOCUMENT?                   304
                   |
      +------------+
      |  NO        |   YES
      |            V
      |     DETERMINE NEW ATTRIBUTES TO BE SENT,      306
      |       LEAVING LOCAL CATALOG ALONE
      |            |
      V            |
      SEND DOCUMENT AND ATTRIBUTES TO TARGET SERVER     308
                   |
                   V
       ACKNOWLEDGMENT OF SUCCESSFUL RECEIPT        312
                   |
                   V
     UPDATE MASTER CATALOG TO RECORD TRANSFER AND NEW
           LOCATION FOR DOCUMENT MASTER                   314
                   |
                   V
                 DONE
```

MIGRATION DETAIL FLOW

FIG. 4

RECEIPT BY TARGET SERVER

DOCUMENT AND SUGGESTED ATTRIBUTES,
FROM THE SOURCE SERVER 402

DOES TARGET SERVER'S POLICY DEFINITION ALLOW
ATTRIBUTES TO BE ACCEPTED FROM SOURCE SERVER? 404

NO

YES

VALIDATE ATTRIBUTES, OVERRIDE ERRORS 406

USE TARGET SERVER'S LOCAL POLICY TO ASSIGN NEW
ATTRIBUTES TO THE INCOMING DOCUMENT 408

STORE DOCUMENT AND ATTRIBUTES IN TARGET SERVER 410

SCHEDULE NEXT MANAGEMENT ACTION FOR DOCUMENT
AT THE TARGET SERVER 412

DOES TARGET SERVER POLICY ALLOW ATTRIBUTES FOR
TIME BASED ON CURRENT DATE AND MANAGEMENT POLICIES
FOR THAT SERVER AND DOCUMENT 414

DONE

RECEIPT BY TARGET SERVER

FIG. 5

RETRIEVAL BY LOCAL SERVER FROM REMOTE SERVER

```
        LOCAL SERVER DETERMINES IT REQUIRES A COPY OF
          A DOCUMENT STORED AT REMOTE SERVER                    502
                     |
                     V
        LOCAL SERVER SENDS MESSAGE TO REMOTE SERVER
      WITH THAT DOCUMENT, ASKING FOR A COPY               504
                     |
                     V
         REMOTE SERVER REPLIES WITH THE DOCUMENT
           AND ITS POLICY ATTRIBUTES                       506
                     |
                     V
      DOES LOCAL SERVER ACCEPT ATTRIBUTES OF COPIES
          FROM REMOTE SERVER?                              508
      +--------------|
      |  NO          |     YES
      |              V                          510
      |   VALIDATE ATTRIBUTES, OVERRIDE ERRORS  ------+
      |                                               |
      V                                               |
      USE LOCAL SERVER'S LOCAL POLICY TO ASSIGN NEW   |
        ATTRIBUTES TO THE INCOMING DOCUMENT           |
                     |                     512        |
                     |--------------------------------+
                     V
    STORE DOCUMENT AND ATTRIBUTES IN LOCAL SERVER    514
          AS A DOCUMENT COPY
                     |
                     V
     SCHEDULE NEXT MANAGEMENT ACTION FOR DOCUMENT    516
            AT THE LOCAL SERVER
                     |
                     V
                    DONE
```

RETRIEVAL FROM REMOTE SERVER

FIGURE 6A

DATE = 030393: INPUT JONES DOCUMENT

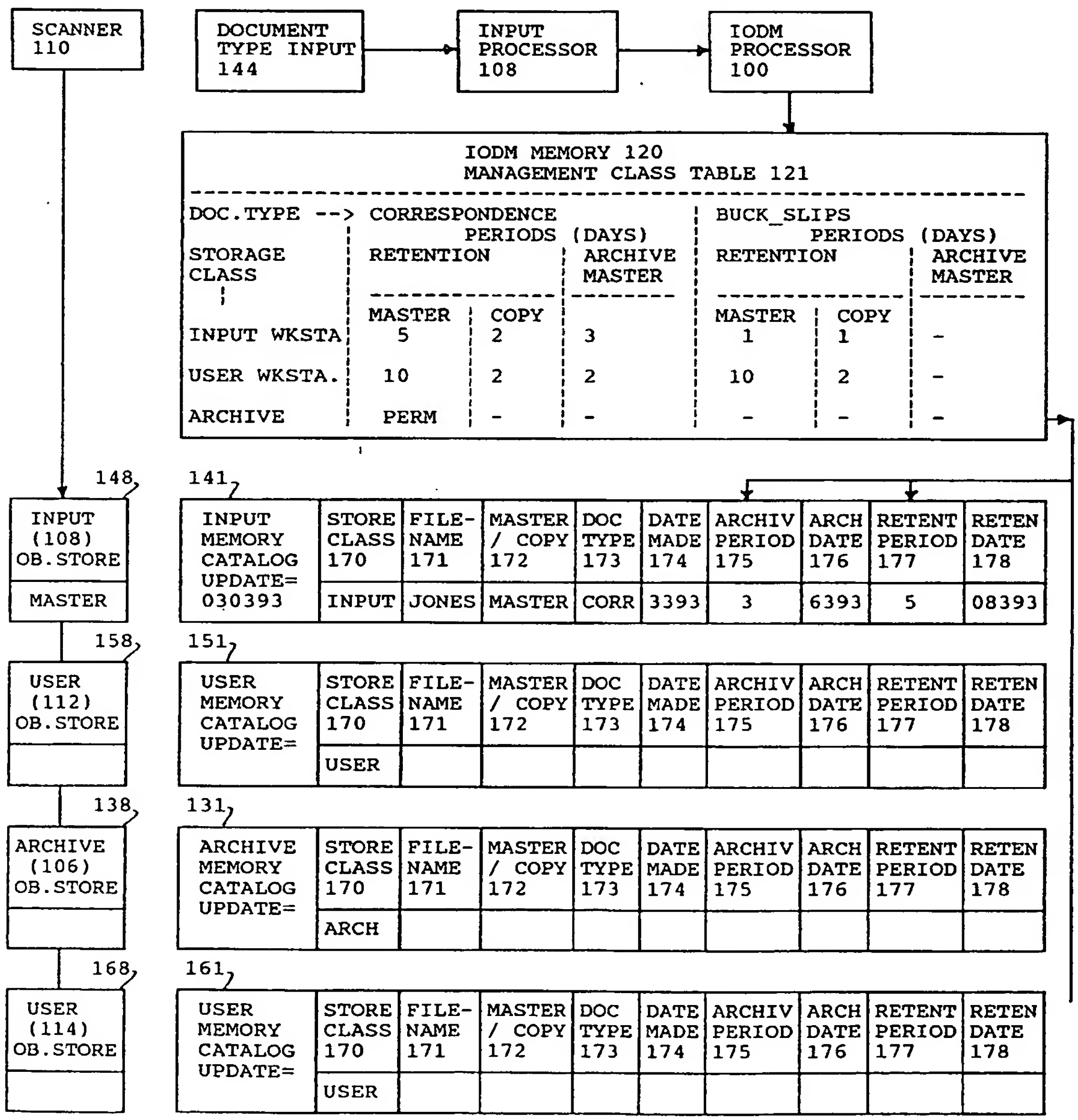
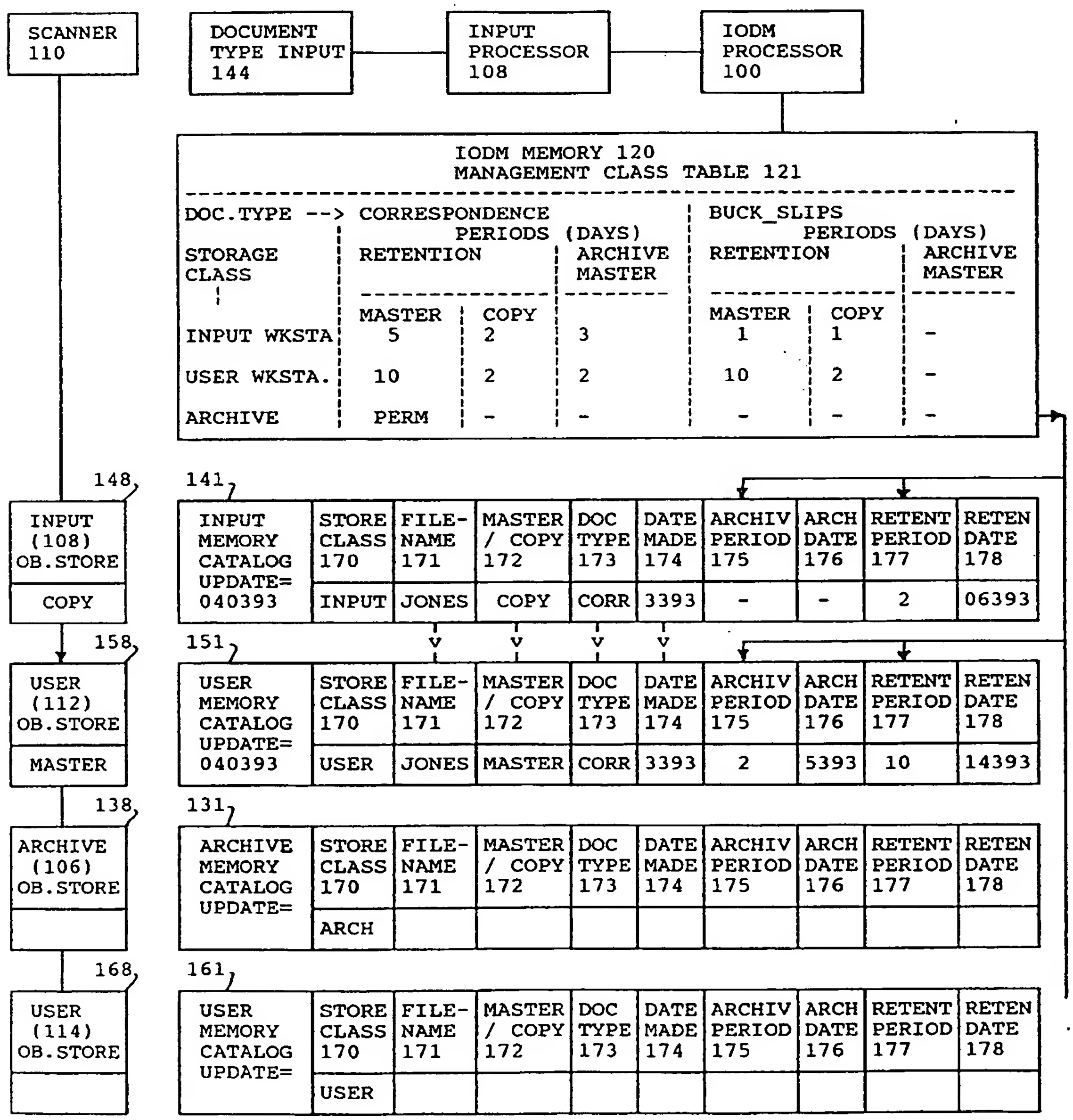


FIGURE 6B DATE = 040393: SEND MASTER TO USER WORKSTATION 112
KEEP COPY AT INPUT WORKSTATION 108



IODM MEMORY 120
MANAGEMENT CLASS TABLE 121

DOC.TYPE --> STORAGE CLASS	CORRESPONDENCE PERIODS (DAYS) RETENTION MASTER	COPY	ARCHIVE MASTER	BUCK_SLIPS PERIODS (DAYS) RETENTION MASTER	COPY	ARCHIVE MASTER
INPUT WKSTA	5	2	3	1	1	-
USER WKSTA.	10	2	2	10	2	-
ARCHIVE	PERM	-	-	-	-	-

INPUT MEMORY CATALOG UPDATE= 040393	STORE CLASS 170	FILE-NAME 171	MASTER / COPY 172	DOC TYPE 173	DATE MADE 174	ARCHIV PERIOD 175	ARCH DATE 176	RETENT PERIOD 177	RETEN DATE 178
	INPUT	JONES	COPY	CORR	3393	-	-	2	06393

USER MEMORY CATALOG UPDATE= 040393	STORE CLASS 170	FILE-NAME 171	MASTER / COPY 172	DOC TYPE 173	DATE MADE 174	ARCHIV PERIOD 175	ARCH DATE 176	RETENT PERIOD 177	RETEN DATE 178
	USER	JONES	MASTER	CORR	3393	2	5393	10	14393

ARCHIVE MEMORY CATALOG UPDATE=	STORE CLASS 170	FILE-NAME 171	MASTER / COPY 172	DOC TYPE 173	DATE MADE 174	ARCHIV PERIOD 175	ARCH DATE 176	RETENT PERIOD 177	RETEN DATE 178
	ARCH								

USER MEMORY CATALOG UPDATE=	STORE CLASS 170	FILE-NAME 171	MASTER / COPY 172	DOC TYPE 173	DATE MADE 174	ARCHIV PERIOD 175	ARCH DATE 176	RETENT PERIOD 177	RETEN DATE 178
	USER								

FIGURE 6C

DATE = 060393: ARCHIVE MASTER AT ARCHIVE 106
KEEP COPY AT USER WORKSTATION 112
DELETE COPY AT INPUT WORKSTATION 108

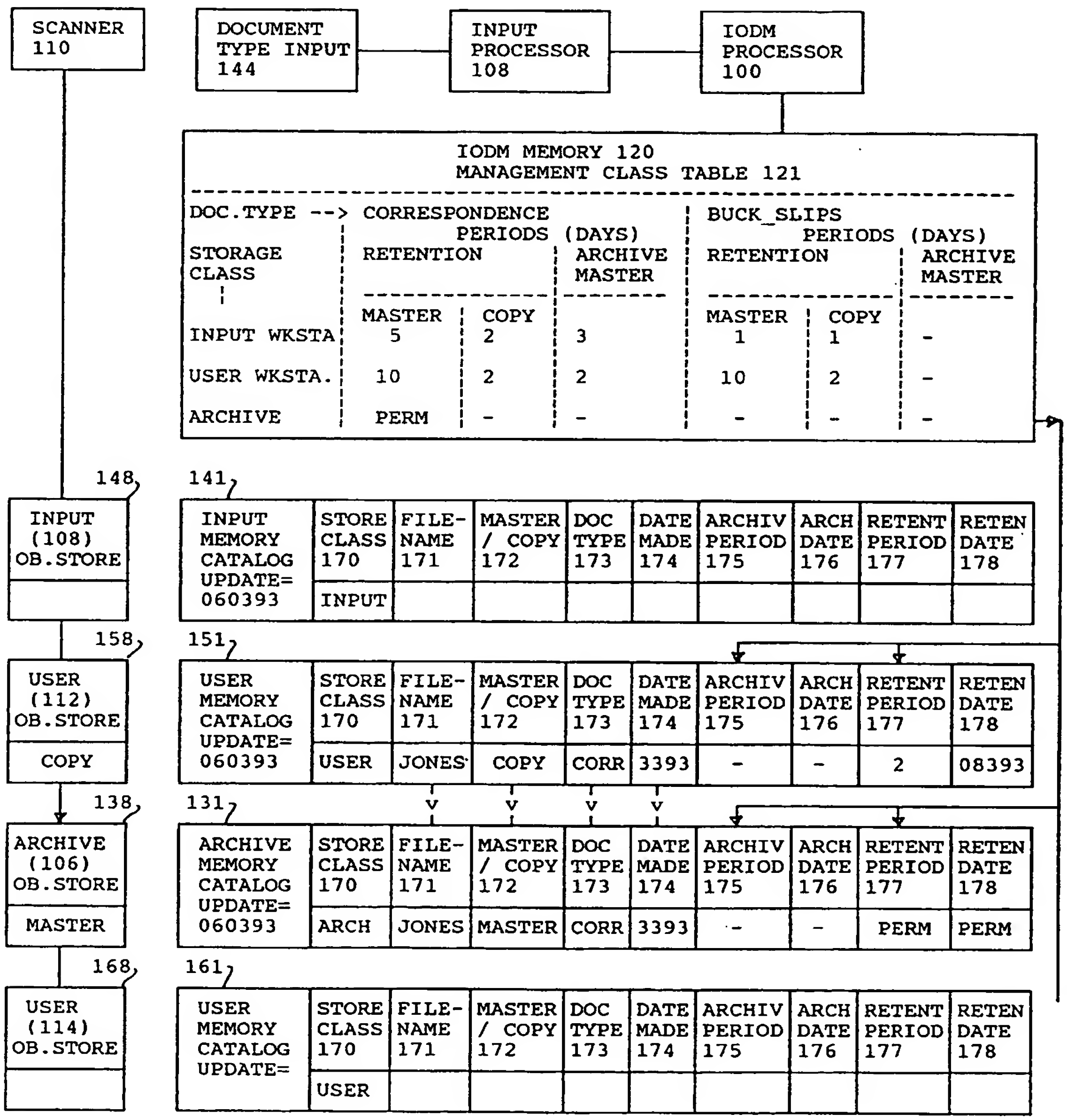


FIGURE 6D

DATE = 070393: SEND COPY TO USER WORKSTATION 114

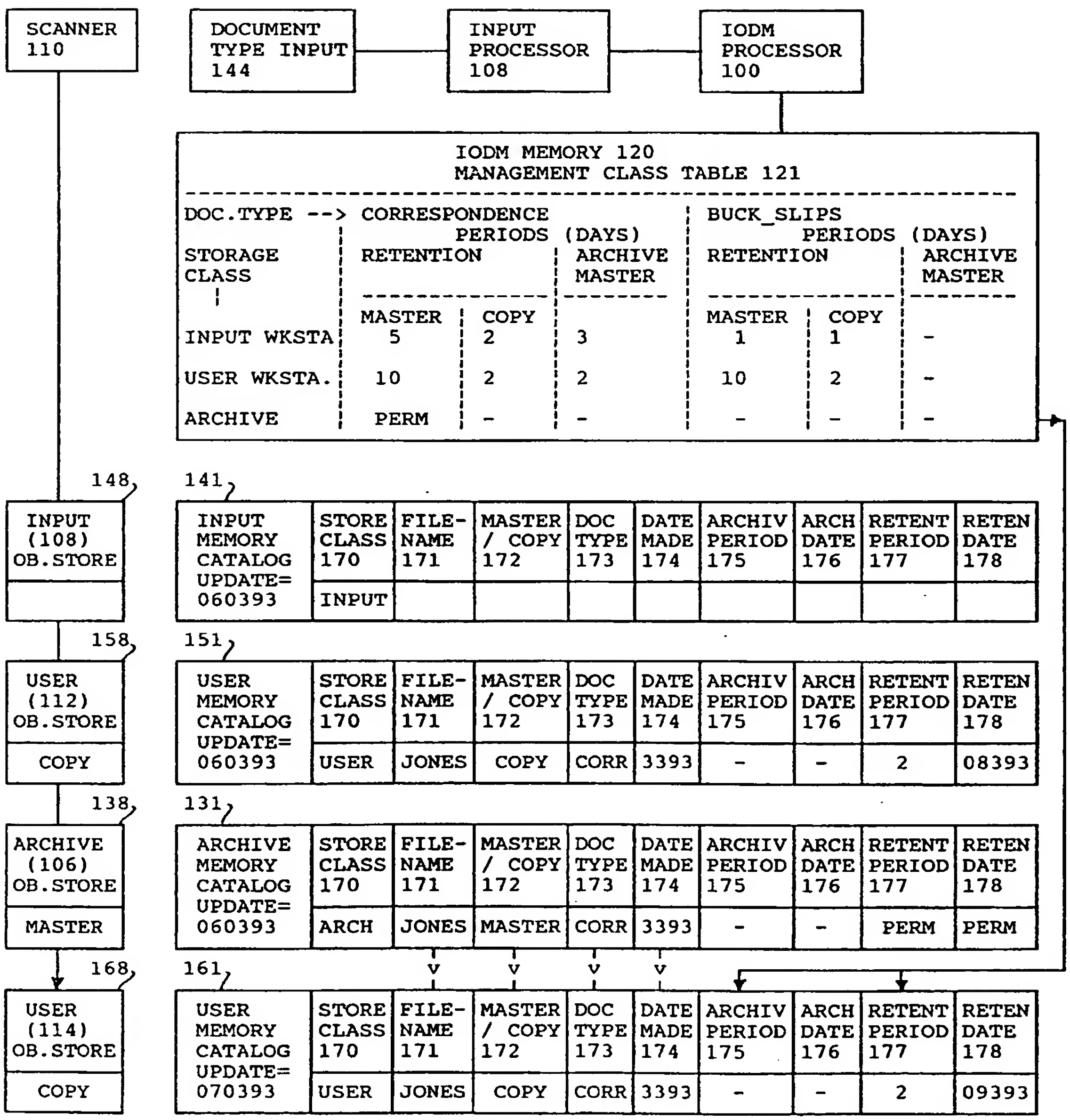


FIGURE 6E

DATE = 050393: SEND MASTER TO USER WORKSTATION 114
KEEP COPY AT USER WORKSTATION 112

SCANNER 110

DOCUMENT TYPE INPUT 144 — INPUT PROCESSOR 108 — IODM PROCESSOR 100

IODM MEMORY 120
MANAGEMENT CLASS TABLE 121

DOC.TYPE --> STORAGE CLASS	CORRESPONDENCE PERIODS (DAYS) RETENTION MASTER	CORRESPONDENCE RETENTION COPY	CORRESPONDENCE ARCHIVE MASTER	BUCK_SLIPS PERIODS (DAYS) RETENTION MASTER	BUCK_SLIPS RETENTION COPY	BUCK_SLIPS ARCHIVE MASTER
INPUT WKSTA	5	2	3	1	1	-
USER WKSTA.	10	2	2	10	2	-
ARCHIVE	PERM	-	-	-	-	-

148 INPUT (108) OB.STORE — COPY

141 INPUT MEMORY CATALOG UPDATE= 040393

STORE CLASS 170	FILE-NAME 171	MASTER / COPY 172	DOC TYPE 173	DATE MADE 174	ARCHIV PERIOD 175	ARCH DATE 176	RETENT PERIOD 177	RETEN DATE 178
INPUT	JONES	COPY	CORR	3393	-	-	2	06393

158 USER (112) OB.STORE — COPY

151 USER MEMORY CATALOG UPDATE= 050393

STORE CLASS 170	FILE-NAME 171	MASTER / COPY 172	DOC TYPE 173	DATE MADE 174	ARCHIV PERIOD 175	ARCH DATE 176	RETENT PERIOD 177	RETEN DATE 178
USER	JONES	COPY	CORR	3393	-	-	2	07393

168 USER (114) OB.STORE — MASTER

161 USER MEMORY CATALOG UPDATE= 050393

STORE CLASS 170	FILE-NAME 171	MASTER / COPY 172	DOC TYPE 173	DATE MADE 174	ARCHIV PERIOD 175	ARCH DATE 176	RETENT PERIOD 177	RETEN DATE 178
USER	JONES	MASTER	CORR	3393	2	5393	10	15393

138 ARCHIVE (106) OB.STORE

131 ARCHIVE MEMORY CATALOG UPDATE=

STORE CLASS 170	FILE-NAME 171	MASTER / COPY 172	DOC TYPE 173	DATE MADE 174	ARCHIV PERIOD 175	ARCH DATE 176	RETENT PERIOD 177	RETEN DATE 178
ARCH								

FIGURE 6F DATE = 050393 (LATER): ARCHIVE MASTER TO ARCHIVE 106
KEEP COPY AT USER WORKSTATION 114

SCANNER 110

DOCUMENT TYPE INPUT 144 — INPUT PROCESSOR 108 — IODM PROCESSOR 100

IODM MEMORY 120
MANAGEMENT CLASS TABLE 121

DOC.TYPE -->	CORRESPONDENCE PERIODS (DAYS)			BUCK_SLIPS PERIODS (DAYS)		
STORAGE CLASS	RETENTION		ARCHIVE MASTER	RETENTION		ARCHIVE MASTER
	MASTER	COPY		MASTER	COPY	
INPUT WKSTA	5	2	3	1	1	-
USER WKSTA.	10	2	2	10	2	-
ARCHIVE	PERM	-	-	-	-	-

148

INPUT (108) OB.STORE

COPY

141

INPUT MEMORY CATALOG UPDATE= 040393	STORE CLASS 170	FILE-NAME 171	MASTER / COPY 172	DOC TYPE 173	DATE MADE 174	ARCHIV PERIOD 175	ARCH DATE 176	RETENT PERIOD 177	RETEN DATE 178
	INPUT	JONES	COPY	CORR	3393	-	-	2	06393

158

USER (112) OB.STORE

COPY

151

USER MEMORY CATALOG UPDATE= 050393	STORE CLASS 170	FILE-NAME 171	MASTER / COPY 172	DOC TYPE 173	DATE MADE 174	ARCHIV PERIOD 175	ARCH DATE 176	RETENT PERIOD 177	RETEN DATE 178
	USER	JONES	COPY	CORR	3393	-	-	2	07393

168

USER (114) OB.STORE

COPY

161

USER MEMORY CATALOG UPDATE= 050393	STORE CLASS 170	FILE-NAME 171	MASTER / COPY 172	DOC TYPE 173	DATE MADE 174	ARCHIV PERIOD 175	ARCH DATE 176	RETENT PERIOD 177	RETEN DATE 178
	USER	JONES	COPY	CORR	3393	-	-	2	07393

138

ARCHIVE (106) OB.STORE

MASTER

131

ARCHIVE MEMORY CATALOG UPDATE= 050393	STORE CLASS 170	FILE-NAME 171	MASTER / COPY 172	DOC TYPE 173	DATE MADE 174	ARCHIV PERIOD 175	ARCH DATE 176	RETENT PERIOD 177	RETEN DATE 178
	ARCH	JONES	MASTER	CORR	3393	-	-	PERM	PERM

(12) UK Patent Application (19) GB (11) 2 367 163 (13) A

(43) Date of A Publication 27.03.2002

(21) Application No 0109568.6

(22) Date of Filing 19.04.2001

(30) Priority Data
(31) 09571673 (32) 15.05.2000 (33) US

(71) Applicant(s)
International Business Machines Corporation
(Incorporated in USA - New York)
Armonk, New York 10504, United States of America

(72) Inventor(s)
Howard Newton Martin
Colin Scott Dawson
David Maxwell Cannon

(74) Agent and/or Address for Service
P Waldner
IBM United Kingdom Limited, Intellectual Property Department, Hursley Park, WINCHESTER, Hampshire, SO21 2JN, United Kingdom

(51) INT CL[7]
G03F 3/06

(52) UK CL (Edition T)
G4A AMT3

(56) Documents Cited
GB 2323945 A EP 0675490 A1

(58) Field of Search
UK CL (Edition T) G4A AFGN AMT3
INT CL[7] G06F 3/06 7/00
ONLINE: WPI, EPODOC, JAPIO, INSPEC

(54) Abstract Title
OPTIMIZED SELECTION AND ACCESSING OF STORED FILES

(57) A method, apparatus, and computer program are disclosed for a computer-implemented technique for generating file copies with minimal mounting and positioning of storage volumes. The method receives a request to generate file copies specifying file selection criteria, identifies matching files meeting the selection criteria, locates the matching files on their storage volumes, and copies the files to a copy set. Determination of file copying order is optimized by placing greater emphasis on relative storage locations of matching files than on the order in which their copies are requested. The method ensures that each matching file is included, without duplication, in the copy set.

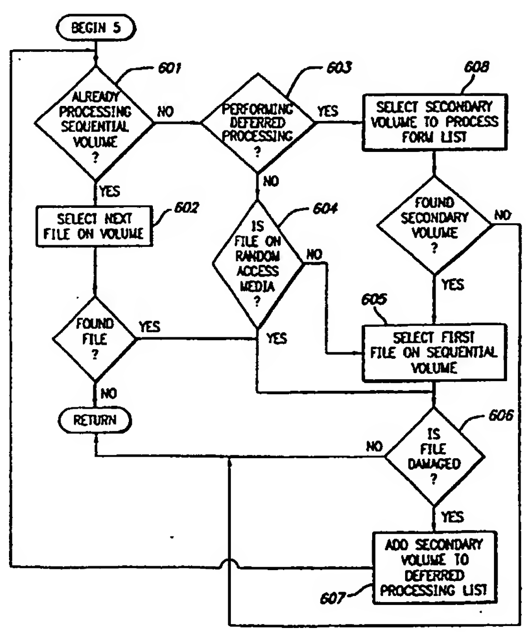
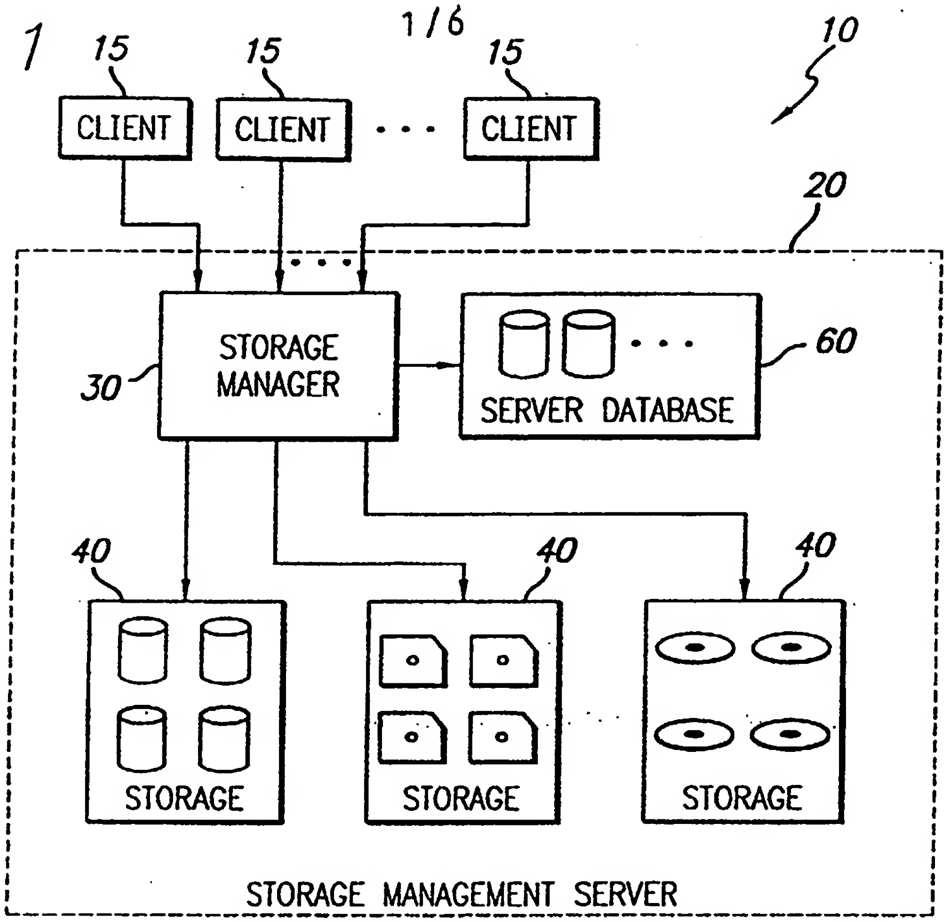


FIG. 7

GB 2 367 163 A

FIG. 1


15
CLIENT
15
CLIENT
15
CLIENT
10
20
STORAGE MANAGER
30
SERVER DATABASE
60
40
STORAGE
40
STORAGE
40
STORAGE
STORAGE MANAGEMENT SERVER

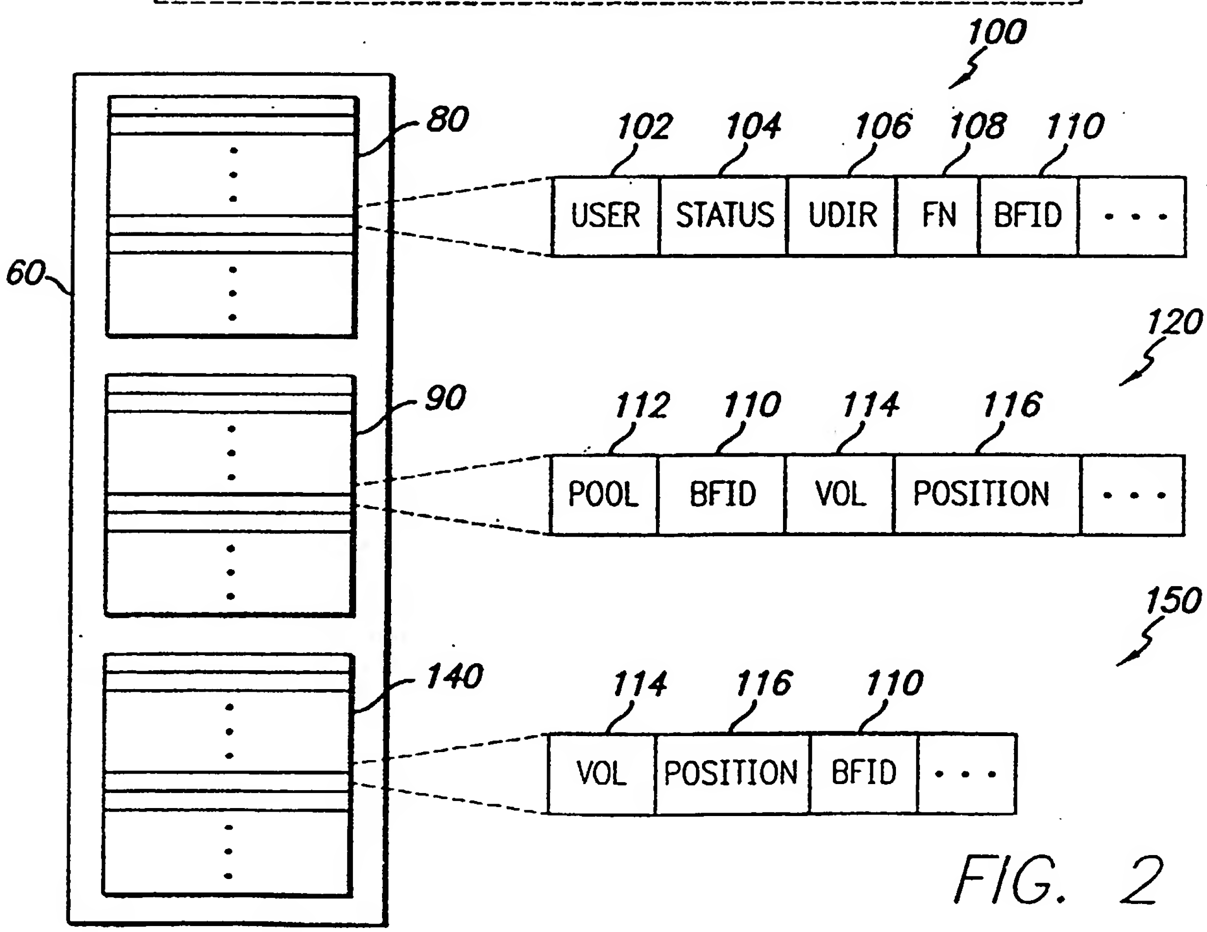

100
80
102
USER
104
STATUS
106
UDIR
108
FN
110
BFID
60
120
90
112
POOL
110
BFID
114
VOL
116
POSITION
150
140
114
VOL
116
POSITION
110
BFID

FIG. 2

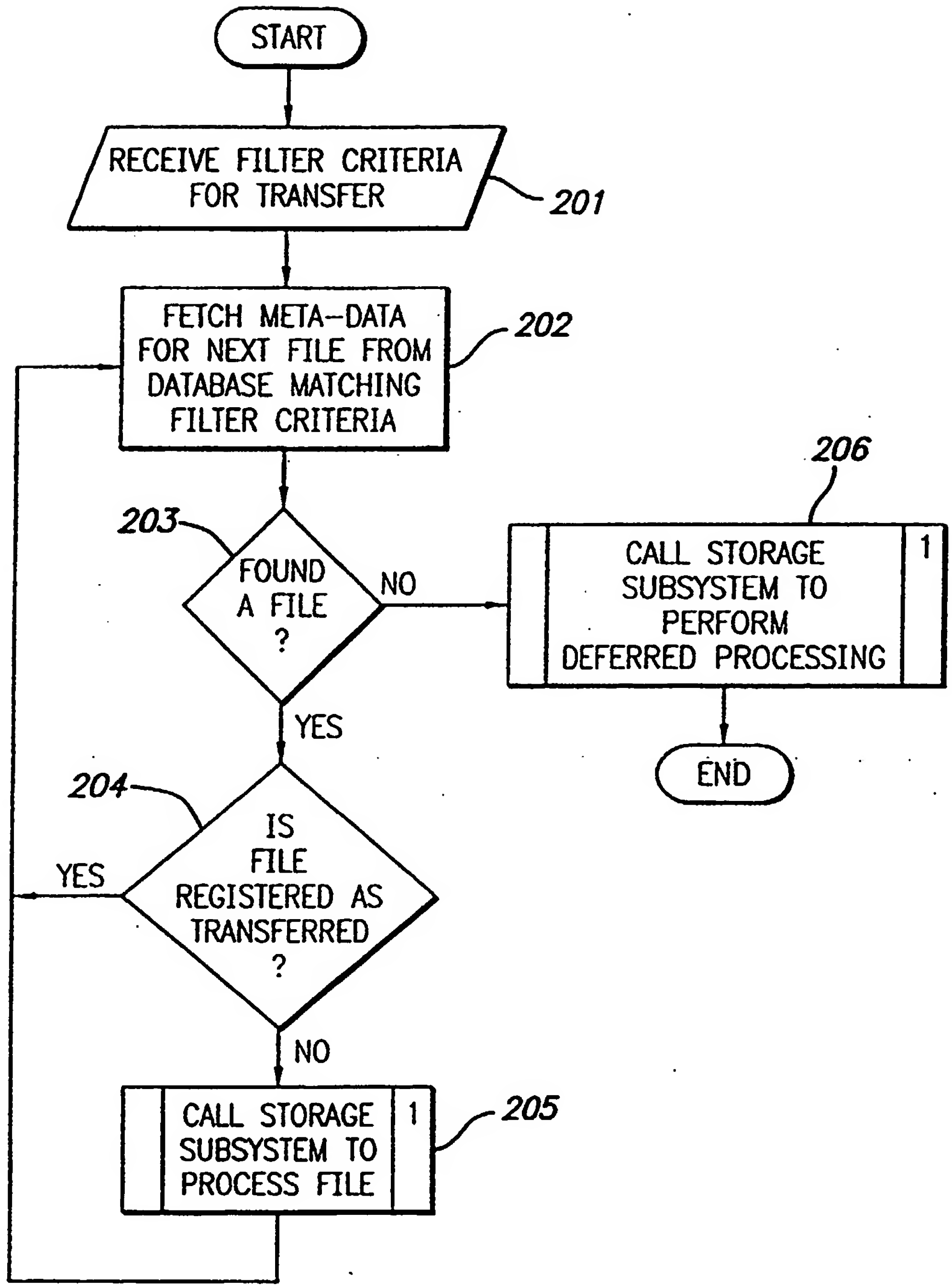

START
RECEIVE FILTER CRITERIA FOR TRANSFER
201
FETCH META-DATA FOR NEXT FILE FROM DATABASE MATCHING FILTER CRITERIA
202
203
FOUND A FILE ?
NO
CALL STORAGE SUBSYSTEM TO PERFORM DEFERRED PROCESSING
1
206
END
YES
204
IS FILE REGISTERED AS TRANSFERRED ?
YES
NO
CALL STORAGE SUBSYSTEM TO PROCESS FILE
1
205

FIG. 3

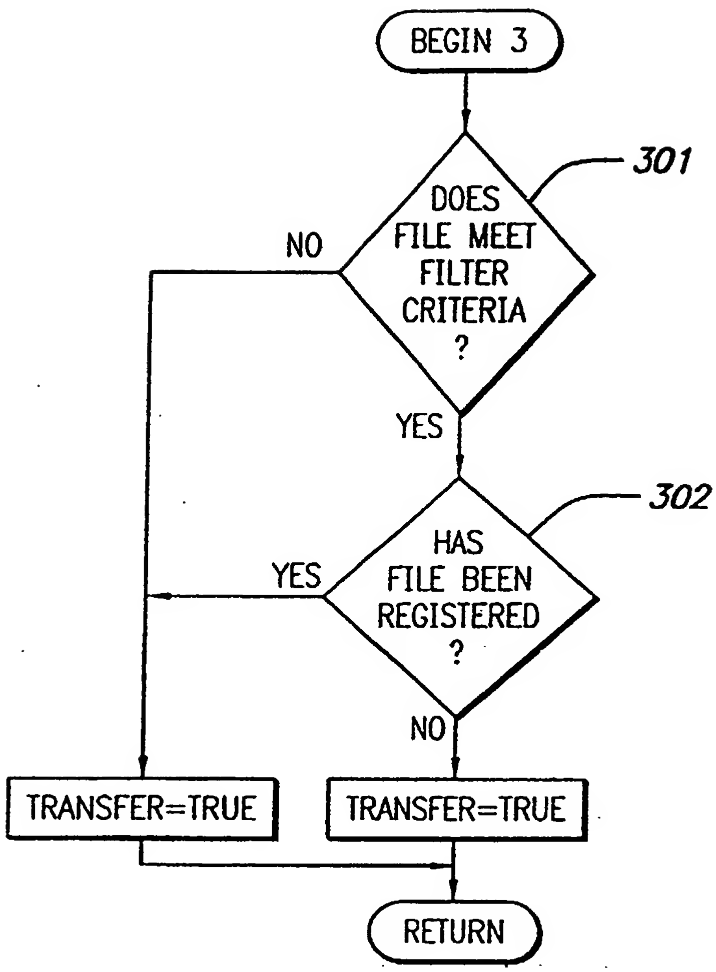

BEGIN 3
301
DOES FILE MEET FILTER CRITERIA ?
NO
YES
302
HAS FILE BEEN REGISTERED ?
YES
NO
TRANSFER=TRUE
TRANSFER=TRUE
RETURN

FIG. 4

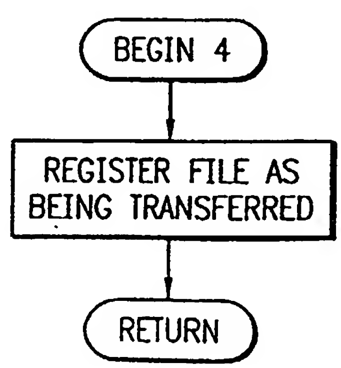
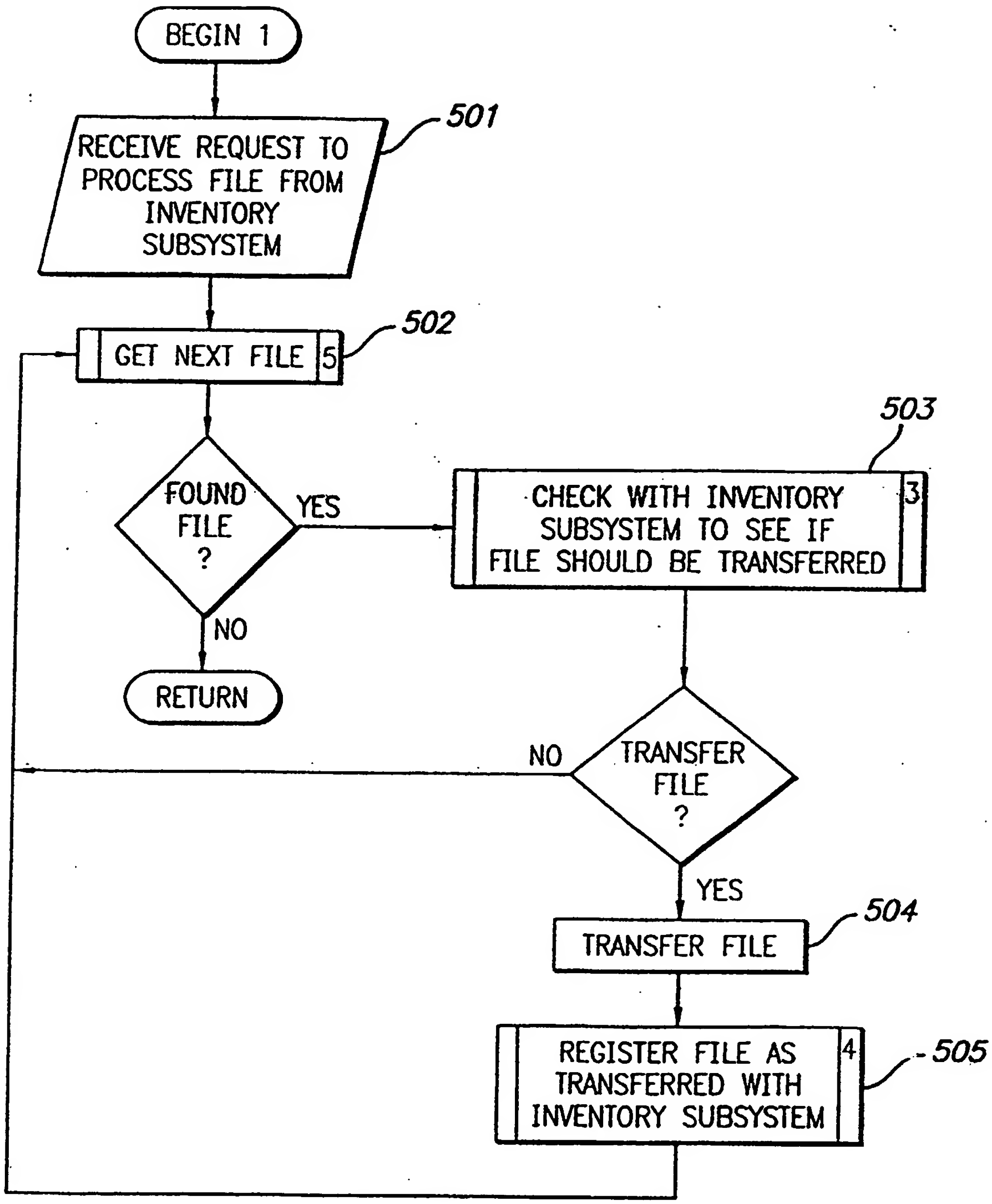

BEGIN 4
REGISTER FILE AS BEING TRANSFERRED
RETURN

FIG. 5



BEGIN 1
RECEIVE REQUEST TO PROCESS FILE FROM INVENTORY SUBSYSTEM
501
GET NEXT FILE
5
502
FOUND FILE ?
YES
NO
RETURN
CHECK WITH INVENTORY SUBSYSTEM TO SEE IF FILE SHOULD BE TRANSFERRED
3
503
TRANSFER FILE ?
NO
YES
TRANSFER FILE
504
REGISTER FILE AS TRANSFERRED WITH INVENTORY SUBSYSTEM
4
505

FIG. 6

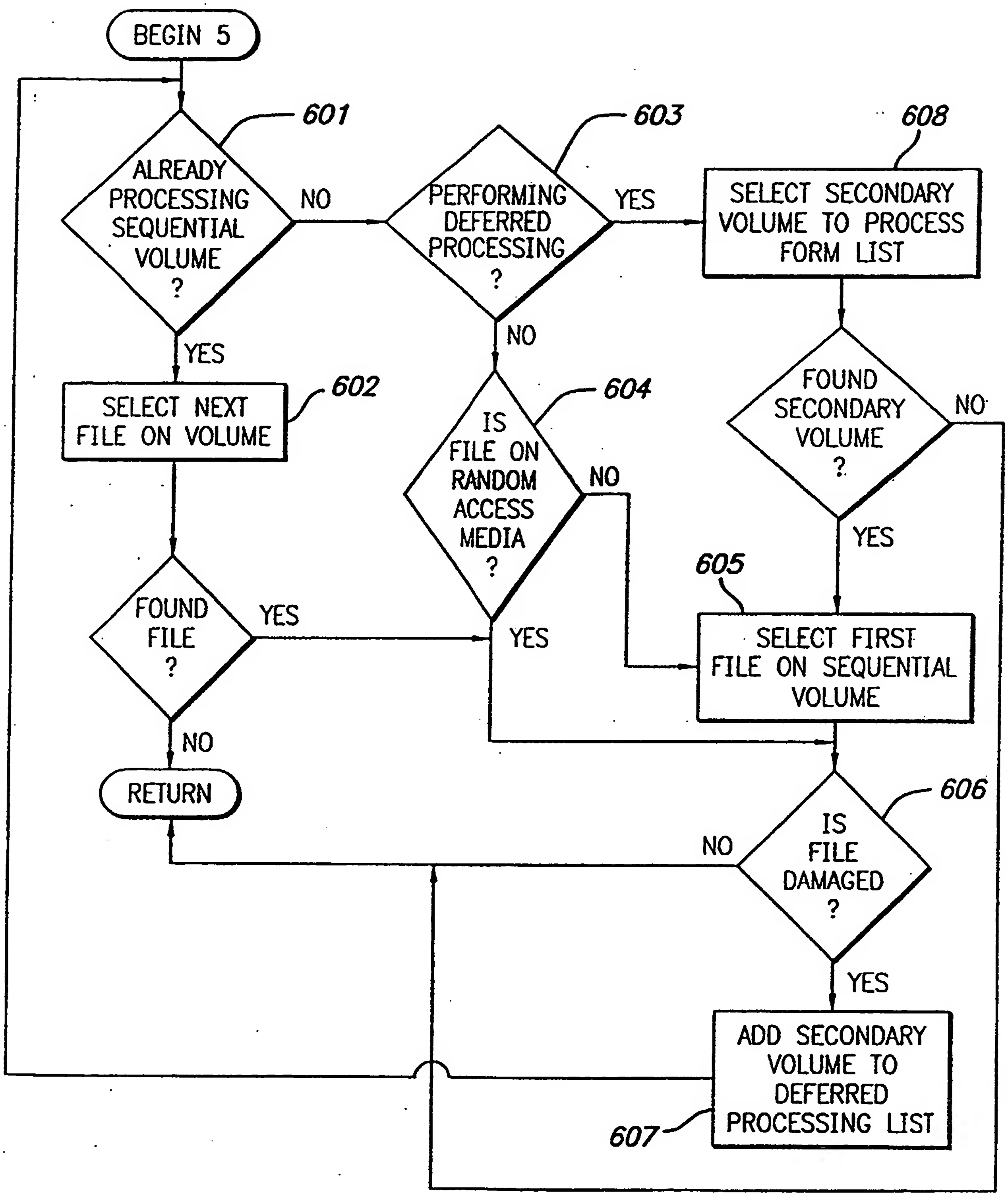

BEGIN 5
601
ALREADY PROCESSING SEQUENTIAL VOLUME ?
NO
YES
603
PERFORMING DEFERRED PROCESSING ?
YES
NO
608
SELECT SECONDARY VOLUME TO PROCESS FORM LIST
602
SELECT NEXT FILE ON VOLUME
604
IS FILE ON RANDOM ACCESS MEDIA ?
NO
YES
FOUND SECONDARY VOLUME ?
NO
YES
FOUND FILE ?
YES
NO
605
SELECT FIRST FILE ON SEQUENTIAL VOLUME
RETURN
606
IS FILE DAMAGED ?
NO
YES
ADD SECONDARY VOLUME TO DEFERRED PROCESSING LIST
607

FIG. 7

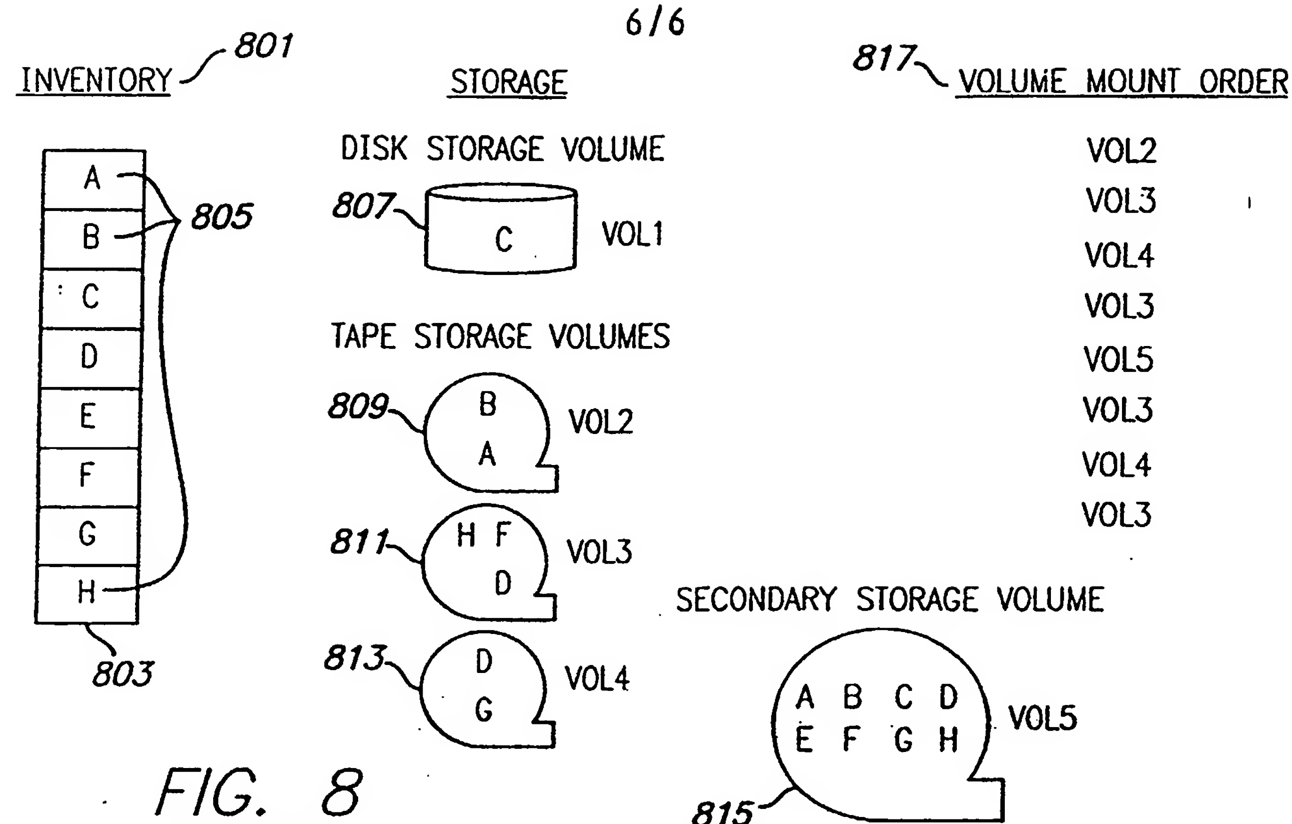


FIG. 8
PRIOR ART

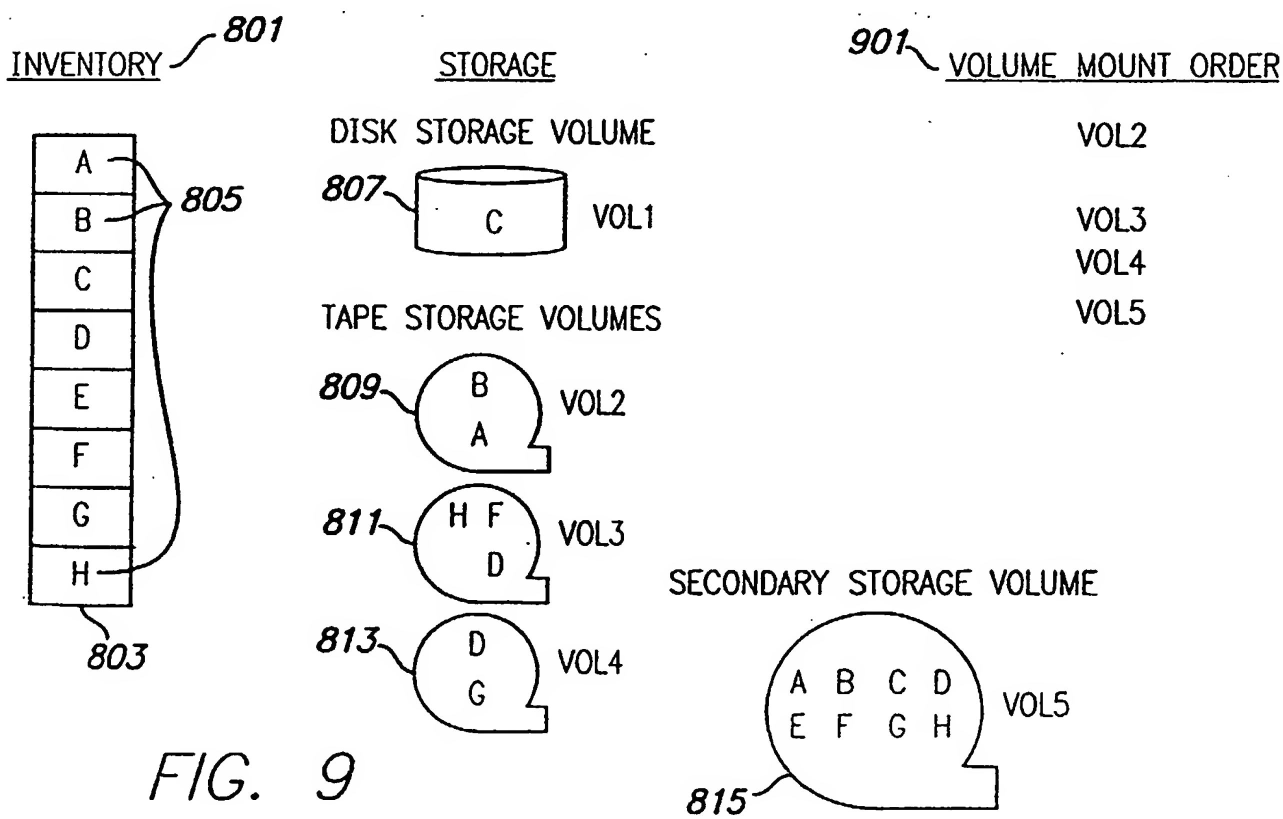


FIG. 9

OPTIMIZED SELECTION AND ACCESSING OF STORED FILES TO AVOID MOUNT AND POSITION THRASHING

The present invention relates generally to data processing systems, and more particularly to storage management servers for optimizing selection and accessing of stored files to avoid mount and position thrashing.

Data processing systems typically require a large amount of data storage. Customer data, or data generated by users within the data processing system, occupy a great portion of this data storage. Effective data processing systems also provide backup copies of this user data to prevent a loss of such data. Many businesses view any loss of data in their data processing systems as catastrophic, severely impacting the success of the business.

A storage management server provides an effective means for protecting customer data. Generally, a client-server configuration includes several clients connected to a single server. The end users create client files and transfer these files to the server. The server receives the client files and stores them on attached storage devices. When used as a storage management system, the server manages the backup, archival, and migration of these client files. By storing the client file on an attached storage device, the server creates a first, or primary, copy of the client file. The server may, in turn, create additional backup copies of the client file for inclusion in the overall storage hierarchy to improve the data availability and data recovery functions of the storage management system. Clients may vary from small personal computer systems to large data processing systems having a host processor connected to several data storage devices. The server can also range from a small personal computer to a large host processor.

An advanced storage management server, such as Tivoli Storage Manager (formerly known as ADSM), maintains reference information about the client files copied within the attached storage volumes. The server uses a database to keep inventory information about the original client files and storage volume location information about the copies of the client files stored within the server. The inventory information typically includes a client system identifier, a client system directory, a client file name, and other attributes of the file. The location information typically consists of a storage volume identifier and a position within the storage volume among other storage attributes. In addition, the server database allows the server to assign a unique identifier to each client file stored

within the attached storage volumes. Thus, the server can track individual files throughout the server storage component.

Accordingly, the server database introduces several advantages to the storage management server. The server can track multiple copies of an individual client file written to different storage volumes. By tracking secondary copies of the client file, the server improves the data availability to the client systems. For example, if a primary copy of a particular client file is inaccessible because it is stored on a destroyed volume or damaged media, the server can access an additional copy residing on a different storage volume and transfer the additional copy to the requesting client system. Further, the server can subsequently recover the unavailable primary copy of the client file from the secondary copy. The server needs both inventory and storage volume location information provided by the server database to accomplish the above-described data recovery.

A data processing system using a storage management server, including a file storage manager, stores files that have been backed up or archived from various client nodes. The server stores client data files in a storage hierarchy consisting of various media types (e.g., magnetic disk, tape, optical disk) and uses a database for tracking the attributes and storage location of each stored client file.

Another function of a storage management server is to select files that satisfy certain criteria, and transfer the files to another location. There are many situations in which the transfer of data to another location is necessary or desirable. For example, it may be desired to create a backup set that represents the latest set of files stored on the server for a particular client node. The backup set could be used for restoring files directly to a client node, without requiring use of a network, or for transporting these files to another server. Those skilled in the art will recognize that the creation of a backup set is only one example and that other applications are well-suited to the copy or transfer of data from one location to another. In general, the specification will refer to copied files as belonging to a copy set.

In this copying operation, data on the source server may be stored on various types of media or volumes. For example, storage media can be removable or non-removable, and can be accessed either sequentially or randomly. Typically, a storage management server can process files from at least three different volume types:. For example, it can process data from random-access, non-removable volumes which do not have to be mounted each time they are accessed and are randomly searched; sequential volumes, such

as tapes, which are mounted at the beginning of the volume and are sequentially processed; and random-access, removable volumes, such as optical disks, which are mounted for each search but randomly processed once mounted.

The description will continue in an illustrative sense with respect to storage volumes, which comprise random-access media and sequential-access media. Random-access media is considered to include media that is both non-removable and random-access. Sequential-access media is considered to include all removable media, whether it is accessed randomly or sequentially.

Information on random-access media, such as magnetic disk, can usually be transferred relatively efficiently. However, transferring data from sequential-access media can impose delays while the required volume is mounted. Moreover, additional delays may be required to position the media to the correct location on the storage volume.

Accordingly, one of the major challenges in generating a copy set or performing any copying operation is to discover how to efficiently copy numerous files from sequential-access media, such as magnetic tapes. The efficient copying should be done with minimal mounting and positioning of input volumes. Therefore, optimized selection and accessing of stored files should avoid mount and position thrashing, which occurs with excessive moving back and forth between the mounted volumes or positions within a volume.

In addition to problems encountered by certain types of media as just described, a further challenge arises from an efficiency problem inherent to the functionality of the copying operation. The problem stems from the utilization of two completely different views of the data, namely the inventory view and the storage view, in the copying operation.

Files are normally selected for inclusion in the copy set based on *inventory view* attributes of the data, important to the end user. Such attributes include the client node, filespace and file name information, and recency of the copy. As used in this specification, the term "filespace" refers to a logical space in the client's storage that can contain a group of files. For example, a filespace could be a logical partition or a directory and its subdirectories.

On the other hand, efficiency requires that files be transferred in some optimal order that depends on the location of these files within the server's storage hierarchy. This information is part of *storage view*

attributes of the data and relates directly to the various types of storage media previously described.

Conventional solutions to the efficiency problem in utilizing both views typically include creating a list of files based on filter criteria which are evaluated using the inventory view. Files in the list are then sorted by their storage location and are transferred in sorted order, which represents the storage view. However, this approach requires a great deal of initial overhead to create and sort the list of files, which delays the transferring of files.

Therefore, there is a need in the art to provide a means whereby a copy set can be generated in optimal manner by considering both the inventory and the storage view of files, and without creation of a sorted list of files.

The present invention discloses a method, apparatus, and computer program for a computer-implemented technique for generating a copy set in such a way as to minimize mounting and positioning of storage volumes. In accordance with the present invention, the method receives a copy set generation request specifying selection criteria for files to be included in a copy set, identifies matching files meeting the selection criteria, locates the matching files on their storage volumes, and copies the files to the copy set, ignoring the file order in the request but considering the proximity of the matching files to each other on the storage volumes. The method ensures that each matching file is included, without duplication, in the copy set, and also ensures that the files are copied with minimal delays in mounting and positioning of the storage volumes.

The storage volumes preferably include sequential-access volumes and random-access volumes, and stored files have a primary copy on a sequential-access volume or a random-access volume, and may have a secondary copy on a sequential-access volume. The copying is preferably attempted in the following order:

1. A request is made to copy a specific file to the target media.

2. If the specified file resides on random-access media, it is copied immediately and no further processing is required for this requested file. At this point, the method gets the next request.

3. Otherwise, if the file is available only on sequential media, the volume is mounted.

4. Once a volume is mounted, the method begins evaluating all files resident on that volume using information stored in the server database to determine which files are eligible for transfer and their positions on the volume. This determination is executed in a position-sensitive (position-optimal) manner so as to minimize positioning within the volume.

5. Each eligible file is copied in position-optimal order.

6. If, while processing a file on the volume, it is determined that the file can not be accessed due to a media defect or hardware failure and if this file also resides on a secondary volume, the secondary volume is added to a list for deferred processing.

7. If the last file on a volume spans to another sequential volume, the first volume is dismounted and the spanned-to, or second, volume is mounted. After mounting the second volume, the method continues processing from step (4) above.

8. Once all eligible files have been transferred from the volume and if the last file on this volume does not span to another volume, the method gets the next request.

9. Once the method has attempted to transfer all eligible files from their primary location, it begins to process any outstanding deferred secondary volumes by transferring all eligible files from each secondary volume in a position-optimal manner. Deferred copying from secondary volumes is handled by the same steps prescribed to primary copying and occurs in a similar preferential-based order.

The apparatus embodiment includes a computer having data storage volumes connected thereto and one or more computer programs, performed by the computer, for executing the above-described method for generating a copy set in such a way as to minimize mounting and positioning of the storage volumes.

A preferred embodiment of the present invention will now be described by way of example only, with reference to the accompanying drawings in which:

FIG. 1 is a block diagram of a data processing system showing a plurality of client systems coupled to a storage management server, according to the preferred embodiments of the present invention;

FIG. 2 is a diagram of a portion of the database included in FIG. 1 showing inventory, reference, and storage volume contents lists, according to the preferred embodiments of the present invention;

FIG. 3 is a flowchart of the inventory component, according to the preferred embodiments of the present invention;

FIG. 4 is a flowchart of the inventory component process 3, according to the preferred embodiments of the present invention;

FIG. 5 is a flowchart of the inventory component process 4, according to the preferred embodiments of the present invention;

FIG. 6 is a flowchart of the storage component, according to the preferred embodiments of the present invention;

FIG. 7 is a flowchart of the storage component process 5, according to the preferred embodiments of the present invention;

FIG. 8 is a diagram showing volume mounting order, according to the prior art; and

FIG. 9 is a diagram showing volume mounting order, according to the preferred embodiments of the present invention.

In the following description of the preferred embodiment, reference is made to the accompanying drawings which form a part hereof, and which is shown by way of illustration of a specific embodiment in which the invention may be practiced.

The preferred embodiments of the present invention provides a means whereby a copy set can be generated in optimal manner by considering both the inventory and the storage view of files. Referring more particularly to FIGS. 1 and 2, like numerals denote like features and structural elements in the various figures. The invention may be as embodied in a data processing system of FIG. 1, using a storage management server to manage one or more copies of client files within the attached storage volumes. In FIG. 1 a data processing system 10 is shown having multiple client systems 15 coupled to a server system 20. The server system 20 includes a storage manager 30 coupled to a server database 60. The storage manager 30 is further coupled to a plurality of storage volumes 40. The storage volumes may consist of various types of storage media, such as magnetic disk, optical disk, or magnetic tape.

Each client system 15 creates original user data files, or client files, which are stored within the corresponding client system 15. The client systems 15 transfer client files to the server system 20. Transferring client files to the server 20 inherently provides a copy mechanism within the server 20 for these original client files. The storage manager 30 directs the client file to an attached storage volume 40. The server 20 stores a first, or primary, copy of the client file on a primary storage volume 40 and may also generate additional copies of the client file on secondary storage volumes 40. The storage manager 30 maintains inventory information about the client file and reference location information pertaining to the copies of the client file within the server database 60.

The server database 60 allows the server 20 to manage individual files within the server storage component 40. The server database 60 introduces advantages to the storage management server 20. The storage manager 30 can track multiple copies of an individual client file written to different storage volumes 40. If the primary copy of a client file is unavailable, the storage manager 30 can access a secondary copy from a different storage volume 40, should a secondary copy be available, using the reference location information in the server database 60. Moreover, the storage manager 30 can recover the primary copy of the client file from a backup copy. In addition, the server database 60 allows the storage manager 30 to coordinate incremental copy operations from a client system 15 to the server 20. The server database 60 denotes which client files have been added to the server storage 40 since a previous incremental copy operation was completed. Without the server database 60, the storage manager 30 must resort to full backup of client data.

FIG. 2 is a diagram showing three portions of the server database 60: a file inventory list 80, a server storage reference list 90, and a storage volume contents list 140. These lists are preferably tables and may also be lists such as linked lists. As stated previously, the server database 60 tracks individual file copies through the server 20. A system utilizing an embodiment of the present invention may include an inventory view that represents user attributes of a file and a storage view that represents storage location.

The inventory view employs a file inventory list 80, shown in FIG. 2, to identify files that match the specified criteria of files to be included in the copy set. An inventory list entry 100 provides inventory information about a client file and facilitates the identification of every file that is found to meet the criteria. Each file that is considered and compared to the criteria has a distinct identifier, denoted the bit-file identifier

(bfid) 110. If the client sends multiple versions of the same file to the server, each of these versions is assigned a distinct bfid 110. Each file that is considered and found to match the criteria is identified by its bfid 110.

An inventory view entry 100 is expanded to show a portion of the inventory information. A server inventory entry 100 typically provides inventory information about the client file. In FIG. 2, a first field contains the user name 102, identifying which client system 15 owns the specified client file. A second field 104 maintains a status indicator 104 for the client file. A third field 106 provides the directory name 106 within the client system 15 where the client file originated. A fourth field 108 contains the file name 108 of the client file. Finally, a fifth field 110 contains the unique file identifier, bfid 110. After a file that matches particular criteria is determined and its bfid is identified, the file may be searched for on storage media.

The storage view utilizes a storage reference list 90, which contains various entries 120, and a storage volume contents list 140 containing entries 150. Each storage reference list entry 120 typically provides reference location information about a particular copy of the client file. A server storage 40 20 can be organized into sets of storage volumes 40, called storage pools. Each set, or pool, is homogenous with respect to media type, in that a pool contains only media of the same type. A file may be located within the server storage 40 20 by specifying the storage pool, the storage volume within the storage pool, and the position within the storage volume. The information in the entry 120 may be arranged so as to locate a file associated with a particular bfid as efficiently as possible. Accordingly, the entry 120 may contain information including a storage pool identifier 112, bfid 110, a storage volume identifier 114, and a position 116 within the storage volume, in that order. This is a reasonable exemplary ordering of information in that it directs the search for a file first to a pool 112 according to media type, and then to bfid. It is noted that in addition to providing file identification, the bfid 110 also serves the purpose of mapping the information within the inventory 80 to the reference location information within the reference list 90. Further, if a file spans multiple storage volumes on the server, a separate reference list entry 120 is used for each volume on which the file is stored.

Continuing with the exemplary embodiment of the present invention, once a volume that is known to contain a requested file has been mounted, file searching is directed by the storage volume contents list 140. Information stored in the entries 150 in this list are ordered by location within the volume such that all requested files stored on the volume may be

copied in the order in which they appear, regardless of the order in which they are requested. A reasonable ordering of the information in the entries 150, then, would comprise the sequence: volume 114, position 116, and bfid 110. This ordering of information facilitates the efficient copying of all requested files on the mounted volume and ensures that once the method of the present invention begins processing a sequential or removable volume, all files on that volume are transferred in optimal order. The present invention, therefore, minimizes the mounting and positioning of volumes during acquisition of all the requested files in the inventory.

As supported by the layout of FIG. 2 and its associated description, the embodiments of the present invention provide a technique whereby a copy set can be generated in optimal manner by considering both the inventory and the storage view of files. The end result is that files are selected based on filter criteria of the inventory view, but are transferred without excessive mounting or positioning of volumes, according to the storage view. One advantage of the present invention is that the efficiency is achieved even if files must be accessed from secondary locations, due to media defects or other problems. Another advantage of this approach is that file transfer begins almost immediately, without the overhead of first sorting files according to their storage location.

In the preferred embodiments of the present invention the inventory component identifies the matching files that meet the specified filter criteria and ensures that each matching file is included, without duplication, in the copy set. For each matching file, the inventory component invokes the storage component which locates the file in the storage hierarchy and copies the file to the copy set, in such a way as to minimize mounting and positioning of storage volumes.

The efficiency is supported by the key feature of the present invention, according to which the storage component does not necessarily copy files in the requested order. Instead, the storage component may anticipate a future request and transfer a file based on its proximity to other files in the storage hierarchy, even before it has been requested to do so. Alternatively, the storage component may receive a request to copy a file but defer processing if it is not possible to retrieve the file from its primary location. After the method has attempted to transfer all files from their primary locations it begins processing any deferred secondary volumes. This optimization, based on storage volume selection as well as file position within a storage volume, avoids mounting a secondary volume whenever a file cannot be transferred from the currently mounted primary volume. The method also ensures that files will be transferred from their primary location if it is possible to do so.

The inventory and storage components interact in a such a way as to ensure that every matching file is copied to the copy set, and that file transfer is performed in an optimal manner with regard to mounting and positioning of storage volumes.

According to exemplary embodiments of the present invention, the inventory component flowcharts are presented in FIGS. 3, 4, and 5. A copy set is typically generated upon command from a storage administrator, who specifies the selection criteria for files in the copy set. These criteria, received in step 201 of FIG. 3, include some selection attributes, such as the name of the client node, the filespaces to which files may belong, the type of files to be included (e.g., backup or archive), and a pattern-matching expression for the file names.

The inventory component uses tables or lists in the server's database for locating files which satisfy the filter criteria given by the selection attributes, as shown in step 202. As the copy set is generated, this component also constructs a temporary table or list which contains an entry for each file that has already been copied to the copy set. The temporary table is used to avoid duplicating the same file within the copy set, but can also facilitate construction of a catalog of files in the copy set.

The inventory component scans the server's database tables, such as the file inventory list 80 in FIG. 2, searching for every file that satisfies the specified filter criteria. Depending on the filter criteria and the organization of the database tables, this can usually be done very efficiently. As it encounters each matching file, found in step 203, the inventory component checks its temporary table to see if the file has already been included in the copy set and registered, according to step 204. If not, the inventory component invokes the storage component, in step 205, to request that this file be copied to the copy set. The matching file is specified to the storage component using a unique identifier, the bfid 110, for that file, which is common to both the inventory and storage component views. If no matching file is found in step 203, the storage component is invoked in step 206 to perform deferred processing. Steps 202-205 are repeated for all files from the database matching the filter criteria.

The inventory component of the exemplary embodiment provides two call-back routines that can be invoked from the storage component. The first call-back routine determines, for any specific file found on a storage volume, whether that file should be added to the copy set. An affirmative response is given if and only if the file satisfies the filter criteria (determined by checking database information against the specific

filter criteria) and has not already been added to the copy set (determined by looking up the file in the temporary table). FIG. 4 illustrates the inventory component executing this call-back routine to check whether a file should be transferred. In step 301 it first determines whether the file meets filter criteria and, if so, it checks in step 302 whether the file has already been registered as transferred.

The second call-back routine provides the ability for the storage component to notify the inventory component that it has successfully copied a file to the copy set. Upon notification, the inventory component adds the file to the temporary table to avoid duplication of files in the copy set. FIG. 5 illustrates the inventory component executing a call-back routine that registers a file as transferred with the inventory component.

After the inventory component has identified all matching files and requested that these be added to the copy set, it invokes the storage component one last time to perform any residual processing in step 206 of FIG. 3. This allows the storage component to defer processing of files stored at secondary locations until all other work has been completed.

The storage component provides an entry point that is invoked by the inventory component to request that a specific file be added to the copy set. The storage component does not necessarily satisfy these requests in the order they are received, since that would be inefficient. Instead, the storage component processes the requested file in conjunction with other files that are stored in close proximity.

In view of the problems associated with various types of media discussed in the background herein, there may be considered two classes of media from which data can be transferred according to the present invention. The first class, including media that is either removable, or sequential, or otherwise inefficient for data access and transfer, is handled in a way to minimize delays and thereby achieve efficiency. The second class, including media that is both non-removable and random-access, is handled in another way because data transfer from such media is inherently efficient. The present invention provides a solution to currently-known inefficiencies in the copying of files from media that is sequential-access.

According to the preferred embodiments of the present invention, the storage component flowcharts are represented in FIGS. 6 and 7. In step 501 a request is received from the inventory component to process a file. A routine is invoked in step 502 to get the next file. If no file is found, the storage component returns. If a file is found, in step 503 the

inventory component call-back routine of FIG. 4 is invoked to verify whether the file should be transferred. If not, step 502 is performed to get another file. If the file should be transferred, it is transferred in step 504, the inventory component call-back routine is invoked in step 505 to register the file, and control is returned to step 502 to get the next file.

In step 502 of FIG. 6, for every request to copy a file to the copy set, the storage component iteratively performs the following steps of FIG. 7 until it returns.

In step 601 it is checked whether the storage component has previously begun retrieving files from a sequential-access volume and, if so, in step 602 the next file on that volume is selected. If a file spans into another sequential-access volume, the routine returns and the spanned-into volume becomes the current volume for the next file to be selected on that volume in the next routine run.

If a sequential volume is not currently being processed, in step 603 it is tested whether the method is already performing deferred processing of secondary volumes. If in step 604 it is determined that the file can be accessed on random-access media, it is immediately selected. Otherwise, if the file can be accessed on a primary sequential-access volume, that volume is selected for processing and the first file on that volume is selected in step 605. If the file is damaged, according to step 606, and can only be accessed on a secondary volume, that volume is placed on a list for deferred processing in step 607. Deferred processing of secondary volumes avoids thrashing that would be caused if a secondary volume were immediately mounted and used for transfer of files.

In step 608 for sequential copying from secondary volumes, when the inventory component has requested that residual (deferred) processing be performed, the next deferred secondary volume becomes the current volume and the first file on that volume is selected in step 605.

After performing the processing described above, the storage component returns to the inventory component for a new request.

FIG. 8 is a diagram showing an example of a file inventory list and its associated volume mounting order according to the prior art. In this example, the inventory view 801 is depicted as an inventory list 803 that stores information about database files. Each file has an individual entry 805 such as the file inventory list entry 100 depicted in FIG. 2. The list 801 is ordered relative to file attributes such as the username and

filename. According to prior art methods, the files 805 in the inventory list 801 are requested in the same order in which they appear in the list 801, and are located and copied from storage media in that order as well.

Primary copies of the files are stored on either disk storage volume 807 or tape storage volumes 809 through 813. Secondary copies of the files are stored on separate tape storage volume 815. In this example, as each file 805 in the inventory list 801 is requested, the method transfers the file from disk storage, if possible. If the file is stored only on sequential media, its corresponding volume is mounted and located such that the file may be accessed and copied. Therefore, to access file A, VOL2 809 is mounted and subsequently positioned to retrieve file A at the end of the tape. File B is requested next, requiring the same tape, VOL2 809, to be re-positioned such that file B may be copied from the beginning of that tape. File C is then copied from disk storage 807. File D is requested next, and requires the removal of VOL2 809, the mounting of VOL3 811, location of the first portion of file D at the end of the volume 811, removal of the volume 811 and mounting of VOL4 813, and location of the latter portion of file D on that volume 813. File E then requires removal of VOL4 813, re-mounting of VOL3 811, and re-positioning of VOL3 811 to the location of file E, which is found to be corrupted. At that point, secondary VOL5 815 would be mounted immediately for the retrieval of file E. File F requires VOL3 811 to be remounted and repositioned, and file G induces the removal of VOL3 811 and re-mounting of VOL4 813. Finally, file H is located by removing VOL4 813, re-mounting VOL3 811 and re-locating to the beginning of that tape. The lengthy volume mount order is listed at 817 in FIG. 8. The example detailed in this diagram is exemplary of the inefficiency inherent in the request and copy of copy data according to methods known in the prior art.

The present invention reduces this inefficiency by optimizing the retrieval of the same files on the same storage volumes utilizing an exemplary method of the invention. Referring to FIG. 9, The file inventory list 801 contains the same files in the same order as the previous example. File A requires the mounting of VOL2 809. Before positioning to the very end of the tape 809, however, the method notes the prior incidence of file B and queries the inventory list 801 for future requests of file B. Noting such a future request, file B is immediately copied by the method, and then file A is copied. The next file in the inventory list is file B, which was already copied in the last step. The method then moves to file C, which is copied from disk storage 807. File D requires the removal of VOL2 809 and the mounting of VOL3 811, however, each of files H, F, and E are encountered on the volume 811 before file D. The inventory list 801 is again queried for anticipated requests of these files. They are found to be

future entries in the list 801 and are therefore copied immediately, except for file E which is corrupted and results in VOL5 815 being designated for deferred processing. The end of VOL3 811 is finally reached and the first part of file D is copied. Continuation of file D on another volume, VOL4 813 requires the removal of VOL3 811 and the mounting of VOL4 813. The method then transfers file G, which is copied directly from the currently mounted and properly positioned VOL4 813. The next three entries in the inventory list, files E, F and G, have already been accounted for in previous steps and are therefore skipped. The last file in the list 801 is file H, which has already been copied from the beginning of VOL3 811. Finally, VOL5 815, which was previously designated for deferred processing, must be mounted for the retrieval of file E. At this point, all of the files have been successfully copied. The shortened mount order is shown at 901 in FIG. 9.

The example as handled by the prior art, shown in FIG. 8, requires 8 separate mounting procedures (excluding retrieval from a disk storage volume) 817, while the same example as handled by a method according to the present invention requires only 4 separate mounting procedures 901. It can clearly be seen that the present invention significantly reduces the mounting and positioning required for acquisition and copying of copy data files. As one of the most expensive operations in storage hierarchy methods is volume mounting and positioning, the present invention directly provides for significant time and cost reductions.

The inventory and storage component of the preferred embodiments of the present invention are implemented in the present invention as software programs being part of the storage management server system of FIG. 1. These software programs can be stored on a storage medium for storing executable computer instructions, such as a magnetic diskette, an optical disk cartridge, or a magnetic tape cartridge, or in memories used to store digital representations of executable computer instructions, such as read-only memory (ROM) or programmable memory (PROM).

The foregoing description of the preferred embodiments of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. For example, the invention may be applied to the transfer of all files, and is not limited to only the most recent files or the generation of a copy set. Determination of position-optimal copying order may be in consideration of media format in addition to file position on the media. For example, consideration of file location on serpentine tape would be different than that for sequential

tape, and would be done so as to minimize the number of tape passes. Another modification may involve the invention utilizing a network for file transfer between a client and a server, in contrast to utilization of removable media. Also, those skilled in the art will appreciate that the invention may be utilized with systems involving file aggregation, whereby all matching files would be transferred from within an aggregate once processing of that aggregate has begun.

CLAIMS

1. A method for transferring files stored on multiple separate storage volumes using information from a list in such a way as to minimize mounting and positioning of the storage volumes, the method comprising:

receiving a request specifying selection criteria for selecting a first file to be included in a copy set;

identifying a first matching file meeting the selection criteria;

locating the selected first matching file on at least one of the storage volumes;

identifying any other matching files on the storage volume having the selected first matching file;

determining a copying order of the first matching file and the any other matching files according to the storage volume having the selected first matching file, such choice being in preference to a choice according to an order in which the matching files were identified; and

copying the selected first matching file and the any other matching files from the storage volume to a copy set according to the determined order.

2. A method as claimed in claim 1 wherein the determining is further according to relative locations of all identified matching files on the storage volume having the selected first matching file.

3. A method as claimed in claim 1 wherein copying the first matching file from random-access media is in preference to copying the first matching file from sequential-access media.

4. A method as claimed in claim 1 wherein the determining of a copying order comprises anticipating a future request for a subsequent matching file.

5. A method as claimed in claim 4 wherein the subsequent matching file is copied before it is requested.

6. A method as claimed in claim 1, further comprising ensuring that each matching file is included, without duplication, in the copy set.